REPORT DOCUMENTATION PAGE

Form Approved OMB NO. 0704-0188

1. REPORT DATE (DD-MM-YYYY)	2. REPORT TYPE	3. DATES COVERED (From - To)
18-04-2017	Final Report	1-May-2016 - 31-Jan-2017

4. TITLE AND SUBTITLE
Final Report: High-Resolution Radar Waveforms Based on Randomized Latin Square Sequences

5a. CONTRACT NUMBER
W911NF-16-1-0144

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
611102

6. AUTHORS
Ram M. Narayanan, Travis D. Bufler

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES AND ADDRESSES
Pennsylvania State University
Office of Sponsored Programs
110 Technology Center Building
University Park, PA 16802 -7000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS (ES)
U.S. Army Research Office
P.O. Box 12211
Research Triangle Park, NC 27709-2211

10. SPONSOR/MONITOR'S ACRONYM(S)
ARO

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
68660-EL-II.3

12. DISTRIBUTION AVAILIBILITY STATEMENT
Approved for Public Release; Distribution Unlimited

13. SUPPLEMENTARY NOTES
The views, opinions and/or findings contained in this report are those of the author(s) and should not contrued as an official Department of the Army position, policy or decision, unless so designated by other documentation.

14. ABSTRACT
This report summarizes the study ofSudoku puzzles as applied to radar applications. We examine Sudoku puzzles for frequency hopped sequences with comparison to the well researched Costas frequency codes. Utilizing the backtracking algorithm, numerous Sudoku puzzles varying in size are generated for analysis of their co-hit and cross-hit properties as compared to Latin Squares. We look at how Sudoku sequences compare to traditional waveforms such as linear frequency modulation and rectangular pulse. Furthermore, we investigate the use of Sudoku puzzles for antenna applications including array interleaving, array thinning, and random element spacing

15. SUBJECT TERMS
Sudoku, Costas, Frequency Hopped, Array Thinning, Beamforming, High Resolution, Ambiguity Function

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	15. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Ram Narayanan
UU	UU	UU	UU		19b. TELEPHONE NUMBER 814-863-2602

RPPR

as of 01-Nov-2017

Agency Code:

Proposal Number: **Agreement Number:**

Organization:
Address: , ,
Country:
DUNS Number: EIN:
Report Date: Date Received:
for Period Beginning and Ending
Title:
Begin Performance Period: **End Performance Period:**
Report Term: -
Submitted By: Email:
Phone:

Distribution Statement: -

STEM Degrees: **STEM Participants:**

Major Goals:

Accomplishments:

Training Opportunities:

Results Dissemination:

Plans Next Period:

Honors and Awards:

Protocol Activity Status:

Technology Transfer:

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
31-01-2017	Final	01 May 2016 – 31 January 2017

4. TITLE AND SUBTITLE
High-Resolution Radar Waveforms Based on Randomized Latin Square Sequences

5a. CONTRACT NUMBER
W911NF-16-1-0144

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Ram M. Narayanan and Travis D. Bufler

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
The Pennsylvania State University
201 Old Main
University Park, PA 16802-1505

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
US ARMY ACC-APG-RTP W911NF
4300 S. MIAMI BLVD.
DURHAM NC 27703

10. SPONSOR/MONITOR'S ACRONYM(S)
ARO

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This report summarizes the study of Sudoku puzzles as applied to radar applications. We examine Sudoku puzzles for frequency hopped sequences with comparison to the well researched Costas frequency codes. Utilizing the backtracking algorithm, numerous Sudoku puzzles varying in size are generated for analysis of their co-hit and cross-hit properties as compared to Latin Squares. We look at how Sudoku sequences compare to traditional waveforms such as linear frequency modulation and rectangular pulse. Furthermore, we investigate the use of Sudoku puzzles for antenna applications including array interleaving, array thinning, and random element spacing. Finally, radar simulations and comparisons are performed using the frequency coded Sudoku sequences for stationary and moving targets. It is concluded that Sudoku based sequences and antenna array configurations show significant promise in architecting high-resolution radar waveforms and enhancing antenna beamforming capabilities.

15. SUBJECT TERMS
Sudoku, Costas, Frequency Hopped, Array Thinning, Beamforming, High Resolution, Ambiguity Function

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Ram M. Narayanan
Unclassified	Unclassified	Unclassified	None	71	19b. TELEPHONE NUMBER *(Include area code)* (814) 863-2602

High-Resolution Radar Waveforms Based on Randomized Latin Square Sequences

Abstract

This report summarizes the study of Sudoku puzzles as applied to radar applications. We examine Sudoku puzzles for frequency hopped sequences with comparisons to the well researched Costas frequency codes. Utilizing the backtracking algorithm, numerous Sudoku puzzles varying in size are generated for analysis of their co-hit and cross-hit properties as compared to Latin Squares. We look at how Sudoku sequences compare to traditional waveforms such as linear frequency modulation and rectangular pulse. Furthermore, we investigate the use of Sudoku puzzles for antenna applications including array interleaving, array thinning, and random element spacing. Finally, radar simulations and comparisons are performed using the frequency coded Sudoku sequences for stationary and moving targets. It is concluded that Sudoku based waveforms and antenna array configurations show significant promise in architecting high-resolution radar waveforms and antenna beamforming capabilities.

Table Of Contents

1 Introduction

Sudoku has become an increasing popular puzzle over the last decade and has found its way in many daily newspapers. This report discusses its roots and looks at ways these puzzles can be applied to different radar applications, specifically, high resolution radar. The topics in which we consider are frequency coded waveform design, antenna arrays, and radar target detection. The Summary at the end encourages future Sudoku research into other subjects related to Radar and system designs.

1.1 High Resolution Radar

The definition of high resolution radar is based on the central operating frequency and bandwidth of the radar system. We can use the definition of fractional bandwidth in equation (1), where f_H is the highest used frequency, f_L is the lowest used frequency, and f_c is the center frequency. Depending upon the resulting fractional bandwidth, we can classify our radar operating bandwidth as narrowband, wideband, or ultrawideband [1]. The use of ultra-wideband (UWB) waveforms requires a fractional bandwidth of at least 25%. Considering a radar with a center frequency of 4 GHz, the bandwidth would need to be 1 GHz to be classified as UWB i.e. the f_H would be 4.5 GHz and the f_L would be 3.5 GHz.

$$fBW = \frac{(f_H - f_L)}{f_c} \tag{1}$$

A desirable trait of high resolution radars is having the ability to distinguish closely spaced targets. Utilizing more bandwidth will improve the range resolution, which will provide the radar operator with essential characteristics for the targets and clutter within a given scene. One of the applications we will explore is the use of Sudoku puzzles for frequency coding to reduce ambiguities in the range and Doppler domains. First, we will discuss Sudoku and Latin Squares in the following sections.

1.2 Latin Squares

Latin Squares first examined by Euler are $n \times n$ matrices with n symbols arranged in such a way that every column and row consists of each symbol exactly once. An example of a 3×3 Latin Square is given in Fig. 1 utilizing the symbols A, B, and C. These symbols do not necessarily have to be letters, but could be integers or any other placeholder as long as the row and column constraints are satisfied. it then still qualifies as a valid Latin Square.

We can make some general observations about Latin square arrangements. We can apply basic mathematical operations of rotation, reflection, and exchanging of symbols to produce different, but still valid Latin Squares. Other properties that have been of research interest is Mutually Orthogonal Latin Squares (MOLS) which is defined by taking two Latin Squares of the same order and overlaying them on top of each other

forming $n \times n$ pairs. The resulting Latin squares are considered MOLS if all pairs are distinct. An example of a pair of MOLS is given in Fig. 2 where we see that overlaying these two Latin Squares produces distinct pairs.

A	B	C
B	C	A
C	A	B

(a)

Figure 1: Example Latin Square of order 3.

A	B	C		A	B	C		AA	BB	CC
B	C	A	+	C	A	B	=	BC	CA	AB
C	A	B		B	C	A		CB	AC	BA

(a)

Figure 2: Example Mutually Orthogonal Latin Square

Latin squares have been researched throughout the years for different applications [2]. The main uses of Latin Squares consisted of applying them as error correction codes in which information is coded using Latin Squares to reduce the effects of noise on the transmitted and received data. Experimental designs is another application in which an analysis of variance can be undertaken to assess the affect of multiple variables within experiments.

1.3 Sudoku Puzzles

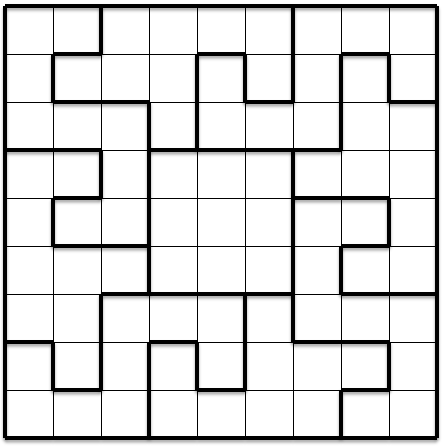
The concept of Sudoku puzzles are an extension and modification of Latin Squares. Sudoku puzzles are combinatorial number-placement arrangement within an $n \times n$ grid (where $n = m^2$) in which each column, each row, and each of the n sub-grids or blocks of size $m \times m$ that compose the grid contains all of the integers from 1 to n. In addition, the same integer appears only once in the same row, column, or any of the n sub-grids of the $n \times n$ grid. The best example is the popular 9×9 Sudoku puzzles in Fig. 3 appearing in daily newspapers and magazines, $n = 9$ and therefore $m = 3$. However, puzzle sizes as large as 121×121 have been realized.

A Sudoku grid is a subset of the well-studied Latin square that meets the additional constraint in that each of its n sub-grids contains the integers from 1 to n [3]. This additional constraint serves to increase its Shannon entropy (measure of the disorder of a typical matrix) as compared to cases without the constraints imposed, i.e. for randomly generated matrices of the same size [4]. Mutually orthogonal Sudoku squares have also been constructed and studied [5, 6]. Sudoku squares are classified as mutually orthogonal if two squares of the same order n can be overlayed on top of one another and produce distinct pairs of numbers. Specifically, if m is prime, a set of $n - m = m(m-1)$ mutually orthogonal $n \times n$ Sudoku squares can be constructed. Thus, there are 6 mutually orthogonal realizations for the popular 9×9 Sudoku puzzle.

$$\left[\begin{array}{ccc|ccc|ccc} 1 & 7 & 2 & 5 & 4 & 9 & 6 & 8 & 3 \\ 6 & 4 & 5 & 8 & 7 & 3 & 2 & 1 & 9 \\ 3 & 8 & 9 & 2 & 6 & 1 & 7 & 4 & 5 \\ \hline 4 & 9 & 6 & 3 & 2 & 7 & 8 & 5 & 1 \\ 8 & 1 & 3 & 4 & 5 & 6 & 9 & 7 & 2 \\ 2 & 5 & 7 & 1 & 9 & 8 & 4 & 3 & 6 \\ \hline 9 & 6 & 4 & 7 & 1 & 5 & 3 & 2 & 8 \\ 7 & 3 & 1 & 6 & 8 & 2 & 5 & 9 & 4 \\ 5 & 2 & 8 & 9 & 3 & 4 & 1 & 6 & 7 \end{array}\right]$$

Figure 3: Example Sudoku puzzle of order 9.

While the square and rectangular sub grids within a Sudoku puzzle are the most common types of puzzles, other geometries exist [7] an example is given in Fig. 4. Here, instead of the integers 1 through N contained within the square or rectangular grids, they have to fit more esoteric shapes.



(a)

Figure 4: Irregular Sudoku puzzle grids in which the symbols or numbers within the grids are no longer restricted to the normal box constraints.

1.4 Sudoku Applications

Just as with Latin Squares, Sudoku puzzles have found similar applications in experimental design, error correcting codes, and cryptography. Research into stenography [8] and image scrambling [9] have also been undertaken. Further research has also been done on Sudoku puzzles themselves where they were examined for their entropy content [10].

2 Ambiguity Functions

The resolution capabilities of a radar system indicate its ability to separate target returns and reject clutter. Resolution is thus an important aspect of the overall system performance, and these considerations often dictate system design philosophy [11]. Targets are resolved in distance and velocity on the basis of combined measurements in range and range rate, or, equivalently, in signal delay and Doppler frequency [12]. It has been shown [13] that the range resolution depends on the signal bandwidth, while the velocity resolution depends on the signal time duration. Furthermore, the combined resolution properties in range and range rate can be described in terms of a two-dimensional ambiguity function which expresses the trade-off between resolutions in both quantities.

The ideal ambiguity function that is most commonly cited is the familiar "thumbtack" surface, which consists of a single spike at the origin, and is zero elsewhere. The single spike eliminates any ambiguities, and its infinitesimal thickness permits resolution of two targets, no matter how closely spaced they might be in range or velocity [14]. Although such a surface is not attainable in practice, several waveform sequences based upon Continuous Discrete Frequency Coding [15] techniques have been developed that approximate the ideal ambiguity diagram while at the same time allowing practical system designs. For such waveforms, any translation of the sequence parallel to the coordinate axes (frequency and time: thus, velocity and range) produces a very low number of out-of-phase coincidences with improved resolution characteristics over other frequency-hopped signals, such as linear frequency hopping sequence [16]. A well-known example is the familiar Costas sequence [17].

The ambiguity function first introduced by Woodward in [13] is used to evaluate the matched filter output of a Radar waveform in the presence of delay and Doppler shift distortions. The radar ambiguity function, $|\chi(\tau, f_d)|$ [1], is defined as

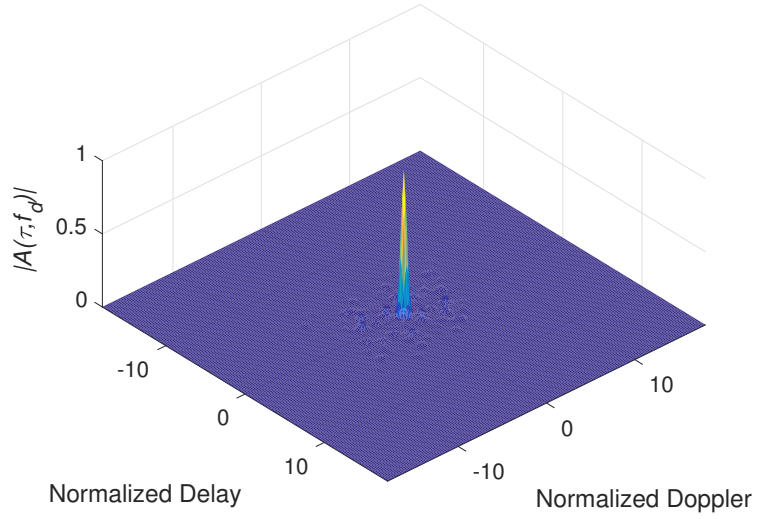
$$|\chi(\tau, f_d)| = \left| \frac{1}{2E} \int_{-\infty}^{\infty} S_1(t) S_2^*(t-\tau) e^{j2\pi f_d t} dt \right|, \tag{2}$$

where S_1 is the transmitted signal, S_2 is the time delayed received signal, τ is the time delay, f_d is the Doppler shift, and E is the energy contained within the signal. Often, equation (2) is termed the auto-ambiguity function if S_1 and S_2 are the same waveform and likewise the cross-ambiguity function if S_1 and S_2 are two different waveforms. The cross-ambiguity is of interest in order to gauge the interference from other waveforms. We know Costas arrays have ideal auto-ambiguity properties but not ideal cross-ambiguity properties and some work has been done on Costas cross-ambiguity analysis [18].

Commonly, the auto-ambiguity function is viewed at two different cuts in the two dimensional ambiguity plane: the first is the zero-Doppler cut $\chi(\tau, 0)$ and the second is the zero-delay cut $\chi(0, f_d)$. A type of ambiguity function that is of interest is a thumbtack like response which is ideally given by a peak at $\chi(0, 0)$ and no ambiguities elsewhere as shown in Fig. 5. Thus far, only approximations to this type of ambiguity function can be

[1] Some authors define the ambiguity function as $|\chi(\tau, f_d)|^2$

achieved. We give an analysis and comparison between several common radar waveforms and examine their properties with respect to Costas and Sudoku coded waveforms.



(a)

Figure 5: "Thumbtack" ambiguity function.

Some of the properties of the ambiguity function are given as follows. The first is symmetry in which $\chi(\tau, f_d) = \chi(-\tau, -f_d)$ and the other property is that the maximum value is located at $\chi(0, 0)$. Additionally, several definitions of the ambiguity function are given in different texts. The definition of the amibguity function that we utilize is $|\chi(\tau, f_d)|$.

2.1 Common Waveform Ambiguity Functions

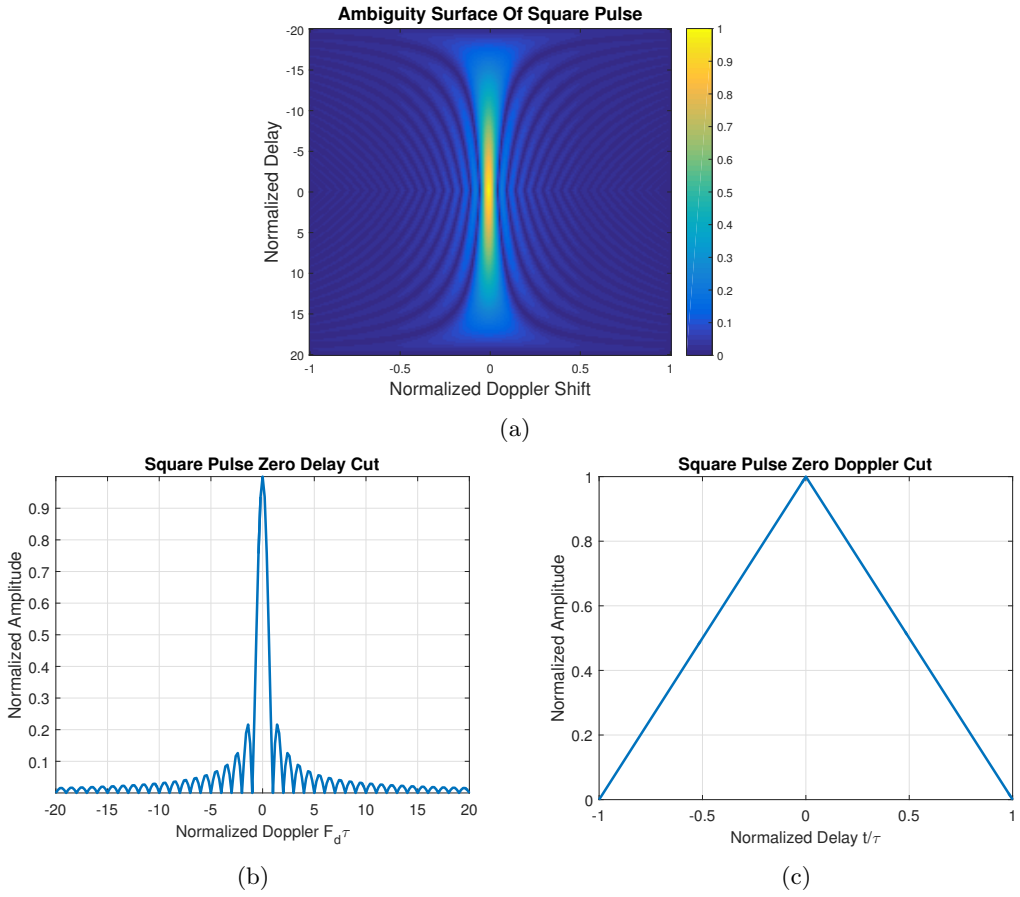
Presented here is a summary and comparison of ambiguity functions between several common waveforms including a rectangular pulse, linear frequency modulation and Costas code to that of the Sudoku coded waveforms.

2.1.1 Rectangular Pulse

The two dimensional auto-ambiguity surface for the rectangular pulse is shown in Fig. 6a and the corresponding delay and Doppler cuts in Fig. 6b and Fig. 6c. The ambiguity function for the rectangular pulse is given as

$$\chi(t, f_d) = \left| \frac{\sin[\pi f_d(\tau - |t|)]}{\tau \pi f_d} \right| \quad -\tau \leq t \leq \tau. \tag{3}$$

where f_d is the Doppler shift, t is the delay shift, and τ is the pulse duration. The ambiguity of a rectangular pulse is spread in the Doppler domain and fairly narrow in the delay direction. We see from the zero-delay cut that the result takes the shape of a sinc function which shows, even for significant Doppler shifts, the matched filter output still produces a high correlation peak. The zero-Doppler cut is the shape of a triangle showing high range ambiguities.



(a)

(b)

(c)

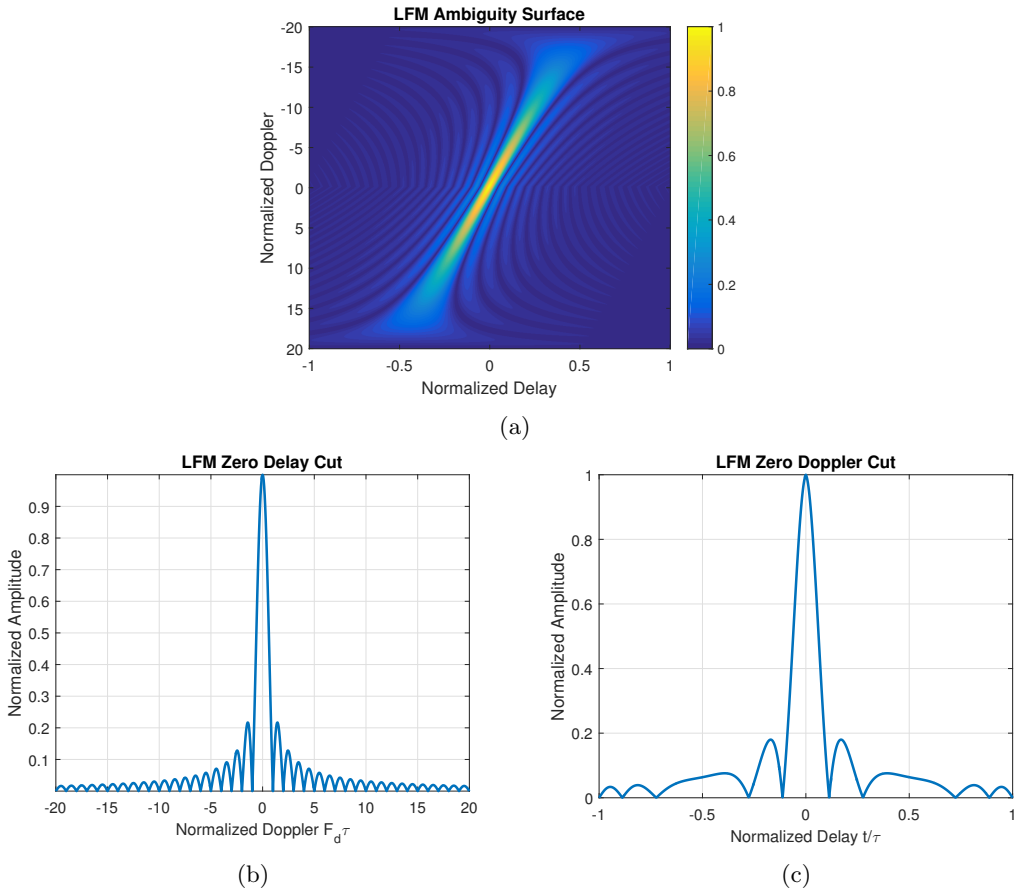
Figure 6: Rectangular pulse auto-ambiguity. (a) Ambiguity surface, (b) zero delay cut, (c) zero Doppler cut.

2.1.2 Linear Frequency Modulation (LFM)

The two dimensional LFM waveform auto-ambiguity surface is shown in Fig. 7a along with the corresponding zero delay and Doppler cuts in Fig. 7b and Fig. 7c. The auto-ambiguity function for the lfm is given as

$$\chi(t, f_d) = \left| \frac{\sin(\pi(f_d + \beta t/\tau)(\tau - |t|))}{\tau\pi(f_d + \beta t/\tau)} \right| \quad -\tau \leq t \leq \tau. \tag{4}$$

where f_d is the Doppler shift, t is the delay shift, β is the bandwidth, and τ is the pulse duration. We observe that the LFM waveform is skewed in the delay and Doppler directions which gives a range-Doppler coupling. A high Doppler shift would result in a large range error for a given target. The LFM zero delay and zero-Doppler cuts are both sinc like. Compared to the the simple pulse, the LFM offers a more narrow peak. The LFM waveform also exhibits sidelobes in range which has the possibility of masking potential targets.



(a)

(b)

(c)

Figure 7: LFM auto-ambiguity. (a) Ambiguity surface, (b) zero-delay cut, (c) zero-Doppler cut.

2.1.3 Frequency Coded Signals

In the case of a frequency-hopped radar, the signal consists of frequencies chosen from a set $\{f_1, f_2, ...f_n\}$ of available frequencies spanning the range $[f_{min}, f_{max}]$, where $B = f_{max} - f_{min}$ is the bandwidth and $\Delta f = B/n$ is the frequency spacing, i.e. $|f_i - f_j| = k\Delta f$ where k is an integer and $1 \leq i, j \leq n$. These frequencies are transmitted at each of a set $\{t_1, t_2, ...t_n\}$ of consecutive time intervals, where $T = t_n - t_1$ is the total signal duration and $\Delta t = t_{k+1} - t_k = T/n$ is the time step or the sub-pulse duration, where $2 \leq k \leq n-1$ [15]. Optimum resolutions in both range and Doppler require, $\Delta f = 1/\Delta t$ [16]. Such a signal can be represented as an $n \times n$ matrix $\mathbf{X}$ shown in Fig. 8 where the n vertical rows correspond to the n frequencies and the n horizontal columns correspond to the n time intervals. The matrix entry x_{pq} equals 1 if and only if frequency f_p is transmitted in time interval t_q; otherwise $x_{pq} = 0$.

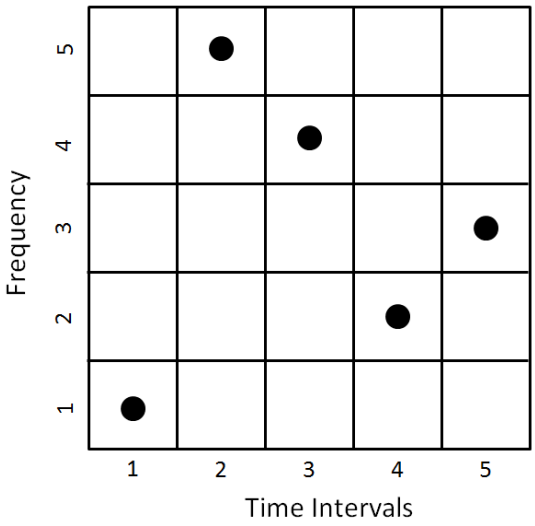


Figure 8: Frequency coded waveform matrix. The column represent the different time intervals and the rows are the individual frequencies.

The auto-ambiguity function for the frequency-hopped radar is expressed in terms of coincidence matrices defined as follows. The auto-coincidence matrix of size $(2n-1) \times (2n-1)$ is denoted as $\boldsymbol{\chi}$ and its elements are obtained from the number of the coincidences of the 1s of the signal matrix $\mathbf{X}$ between the original array and its translation in both coordinate directions [17]. Each entry of $\boldsymbol{\chi}$, denoted by χ_{rs}, represents a shift of r units in the horizontal and s units in the vertical direction. For a shift towards the right, r is positive, while it is negative for a shift towards the left. Similarly, s is positive for a shift upward and negative for a shift downward. The entries of the auto-coincidence matrix $\boldsymbol{\chi}$ satisfy the following conditions: $\chi_{00} = n$ (this is the autocorrelation), $0 \leq \chi_{rs} < n$ for $(r, s) \neq (0, 0)$ when $|r| \leq n-1$ and $|s| \leq n-1$, and $\chi_{rs} = 0$ if $|r| > n$ or if $|s| > n$. For two such signals $\mathbf{X}$ and $\mathbf{Y}$, we define the cross-coincidence matrix $\boldsymbol{\xi}$ of size $(2n-1) \times (2n-1)$ as the number of coincidences of 1s between $\mathbf{X}$ and a translated version of $\mathbf{Y}$, shifted to

the right or left by r and up or down by s . Ideally, we require $\xi_{rs}(\leq n)$ for $|r| \leq n-1$ and $|s| \leq n-1$ to be as close to zero as possible in order to ensure orthogonality between $\mathbf{X}$ and $\mathbf{Y}$. Note that ξ_{00} represents the cross-correlation between $\mathbf{X}$ and $\mathbf{Y}$.

A specific subset of frequency coded waveforms are Costas codes, which give what is known as a "thumbtack" response for the radar ambiguity function. The longer the codes, the closer the approximation is to a true "thumbtack" response. The Costas codes take a $n \times n$ matrix in which the rows are frequencies, and the columns are sub pulses. The matrix is filled in with dots that represent the frequency value associated with a particular subpulse. For a $n \times n$ matrix their are $N!$ ways of filling in the matrix, however if you put the restriction that only one frequency can occupy only one subpulse or time slot then the number of combinations is significantly smaller than $N!$. Another restriction for a correct Costas code is that the distance vectors between all the ones have to be unique.

Fig. 9 shows an example of a valid and invalid Costas code array. While both Fig. 9a and Fig. 9b have placed a 1 only once in a given column and row, Fig. 9b violates the restriction that the distance vectors between all the ones has to be unique by having the distance be the same between adjacent vectors.

$$\begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

(a) Valid Costas code of order 4

$$\begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

(b) Invalid Costas code of order 4

Figure 9: Example of valid and invalid Costas code.

Another way to analyze whether a specific code is a valid Costas code is to compute the difference triangle [19] which is defined as taking the difference between adjacent elements to form $N-1$ rows, where N is the code length. The first row is formed through the difference of immediate adjacent elements, while subsequent rows are formed by taking the next pair of adjacent elements skipping already formed pairs. All entries in a given row must be unique in order for the sequence to be classified as a valid Costas sequence. An example of this process is given in Table 1 for the frequency coded sequence $[1, 3, 4, 2, 5]$. The first entry in row 1 is found as the difference between 3 and 1, while the first entry in row 2 is found through the difference of 4 and 1.

Table 1: Checking a valid Costas Sequence.

Costas	1	3	4	2	5
	2	1	-2	3	
	3	-1	1		
	1	2			
	4				

Table 2 shows this same process repeated for a invalid sequence of $[1, 2, 5, 3, 4]$ where we see repeated elements in row three and it therefore an invalid Costas sequence.

Table 2: Checking an invalid Costas Sequence.

Costas	1	2	5	3	4
	1	3	-2	1	
	4	1	-1		
	2	2			
	3				

The two dimensional Costas auto-ambiguity function is shown in Fig. 10a with the zero-delay and zero-Doppler cuts in Fig. 10b and Fig. 10c for a length 9 sequence. The Costas coded ambiguity function [20] is given in equation (5) which is the sum of the auto $\chi_{nn}(\tau, f_d)$ and cross term $\chi_{nm}(\tau, f_d)$ responses given in equation (6) and equation (7) respectively, as follows,

$$\chi(\tau, f_d) = \chi_{nn}(\tau, f_d) + \chi_{nm}(\tau, f_d), \tag{5}$$

$$\chi_{nn}(\tau, f_d) = \frac{1}{N} \sum_{n=0}^{N-1} e^{j2\pi f_d nT} e^{-j2\pi f_n \tau} e^{j\pi f_d (T+\tau)} \cdot (1 - \frac{\tau}{T}) \frac{\sin[\pi f_d T(1 - \frac{\tau}{T})]}{\pi f_d T(1 - \frac{\tau}{T})}, \tag{6}$$

$$\begin{aligned} \chi_{nm}(\tau, f_d) = & \frac{1}{N(N-1)} \sum_{n=0}^{N-1} e^{j2\pi f_d nT} \sum_{m=0\ m\neq n}^{N-1} e^{-j2\pi f_m[\tau-(n-m)T]} \\ & \cdot e^{-j\pi f_{nmd}(NT+\tau)} (1 - \frac{\tau}{NT}) \frac{\sin[\pi f_{nmd} NT(1 - \frac{\tau}{NT})]}{\pi f_{nmd} NT(1 - \frac{\tau}{NT})}, \end{aligned} \tag{7}$$

where f_d is the Doppler shift, τ is the delay shift, T is the chip length, N is the number of chips, f_m is the m$^{\text{th}}$ frequency code, f_n is the n$^{\text{th}}$ frequency code, and $f_{nmd} = f_n - f_m - f_d$.

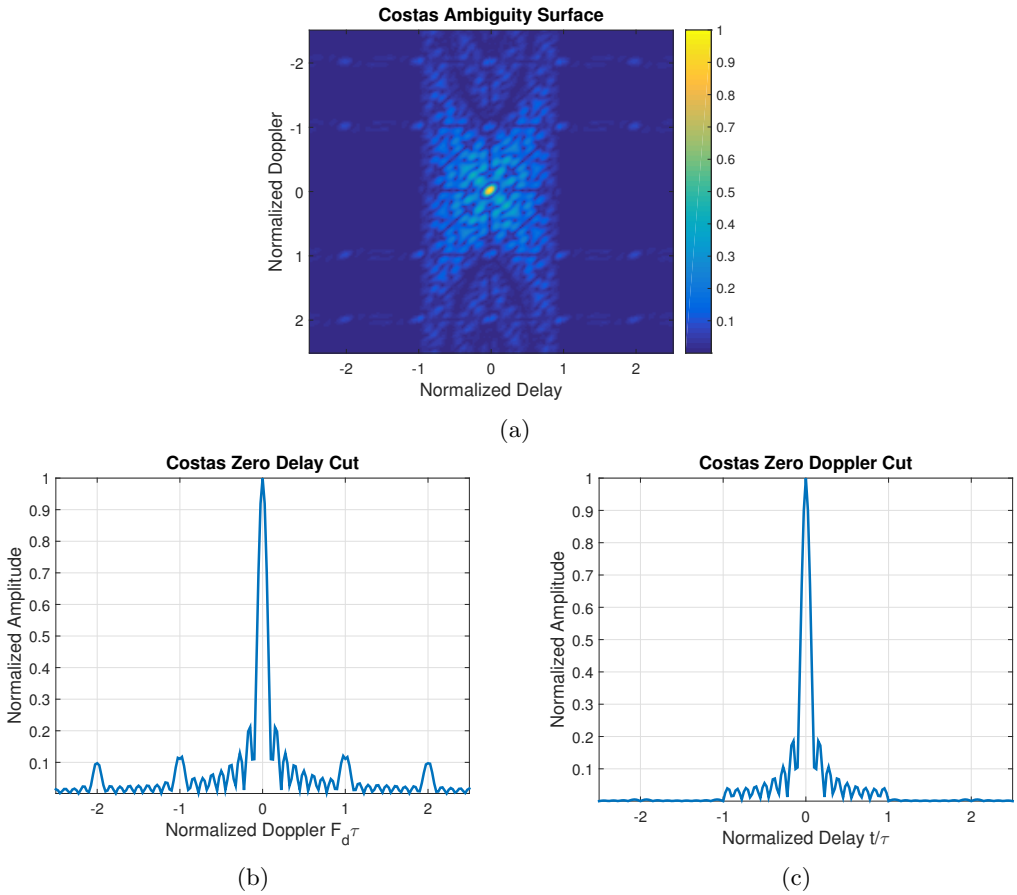


Figure 10: Costas auto-ambiguity plots. (a) Ambiguity surface, (b) zero-delay cut, (c) zero-Doppler cut.

The ambiguity surface plot in Fig. 10a shows a narrow peak at $\chi(0,0)$ with predominantly low side-lobes around the center which can easily be seen from the zero-delay and zero-Doppler cuts. While this is commonly cited as an ideal ambiguity plot, this might not be ideal in the presence of targets with large enough Doppler shifts to essentially make the peak disappear, in which case a different type of ambiguity surface may we desired such as that of the LFM.

2.2 Sudoku Coded Ambiguity Functions

Utilizing Sudoku puzzles for coded waveforms have the potential to offer similar performance to that of the Costas codes with the added benefit of many codes existing compared to the finite amount of Costas codes for a given size. One disadvantage is that the distance vectors may not be unique as in the case for Costas based codes causing more coincidences to occur.

Several constraints are required for a valid Sudoku puzzle which include row, column and box constraints. No number can appear twice in either a given row or column to satisfy the row and column constraints. The box constraint says a number can only appear once with a given box. A representative 9×9 Sudoku puzzle and the same puzzle with only the number "1" is shown in Fig. 11. Using the 1s locations for the order of frequency transmission results in a discrete frequency coded sequence.

1	7	2	5	4	9	6	8	3
6	4	5	8	7	3	2	1	9
3	8	9	2	6	1	7	4	5
4	9	6	3	2	7	8	5	1
8	1	3	4	5	6	9	7	2
2	5	7	1	9	8	4	3	6
9	6	4	7	1	5	3	2	8
7	3	1	6	8	2	5	9	4
5	2	8	9	3	4	1	6	7

(a)

1								
							1	
					1			
								1
	1							
			1					
				1				
		1						
						1		

(b)

Figure 11: Representative Sudoku puzzle. (a) Sudoku puzzle, (b) same puzzle with only the 1s present.

The ambiguity function plot for a Sudoku sequence of length 9 corresponding to the 1s in Fig. 11b is shown in Fig. 12a with the corresponding delay and Doppler cuts in Fig. 12b and 12c. The results are directly comparable to the Costas ambiguity function in which we observe a narrow thumbtack like peak.

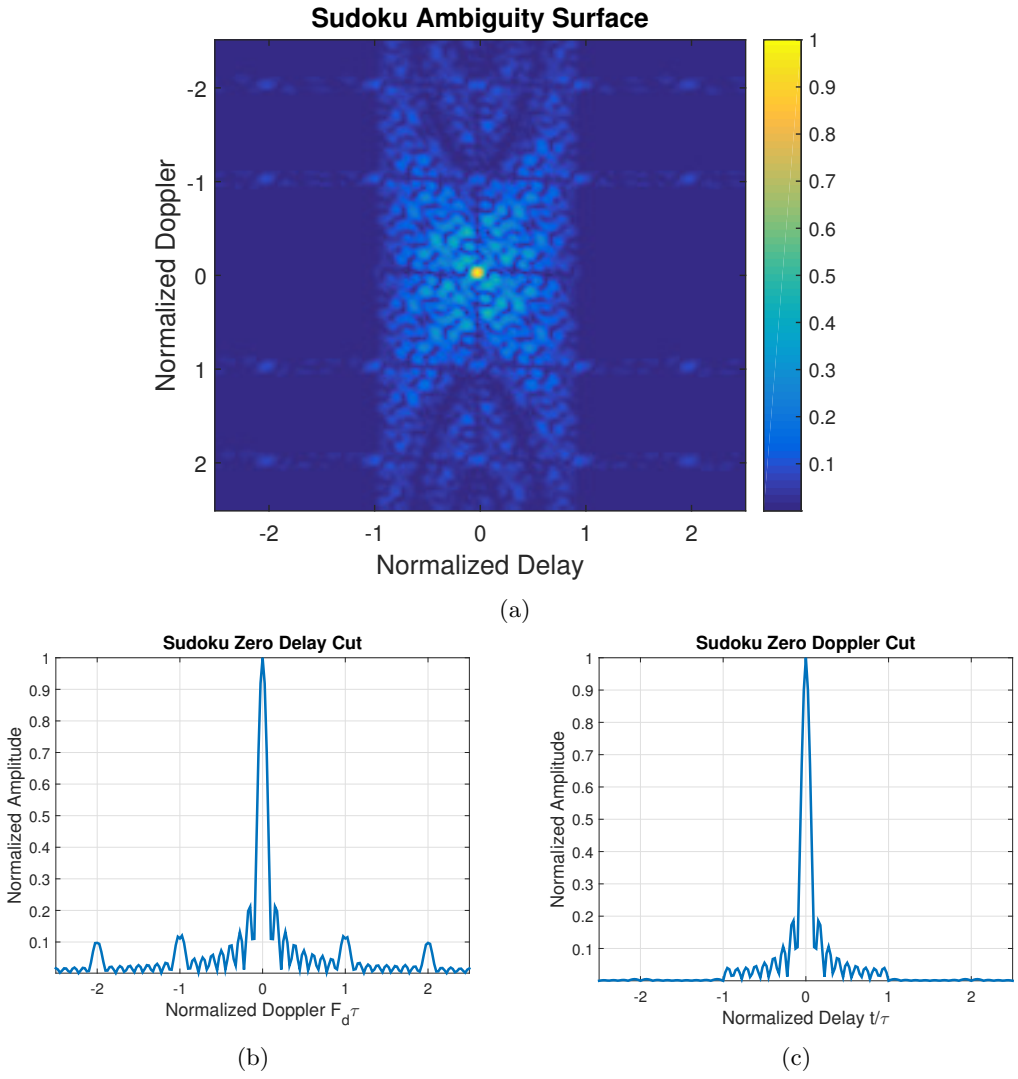


Figure 12: Sudoku auto-ambiguity plots. (a) 2D ambiguity surface (b) zero-delay cut of the Sudoku puzzle, (c) zero-Doppler cut of the Sudoku puzzle.

The results in Fig. 13 and Fig. 14 show the zero-delay and zero-Doppler cuts for the corresponding numbers in the previously described Sudoku puzzle in Fig. 11a. We observe all of the different numbers show a narrow peak in the zero-delay and zero-Doppler cuts. The major advantage of Sudoku coded sequences is the large number of solutions available, unlike Costas codes which only have a finite valid number for a given sequence length.

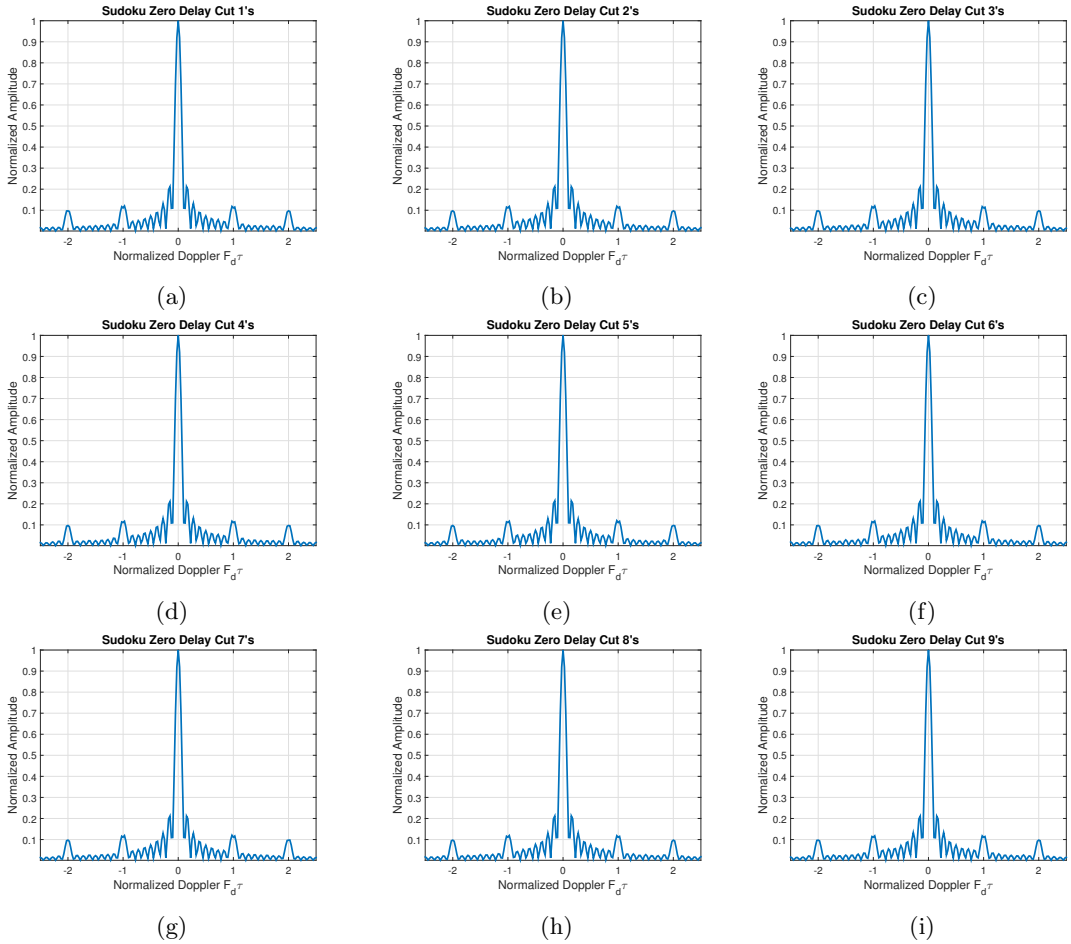


Figure 13: The zero delay cuts for the individual numbers of the previously described Sudoku puzzle solution. (a) number 1s, (b) number 2s, (c) number 3s (d) number 4s, (e) number 5s, (f) number 6s, (g) number 7s, (h) number 8s, (i) number 9s.

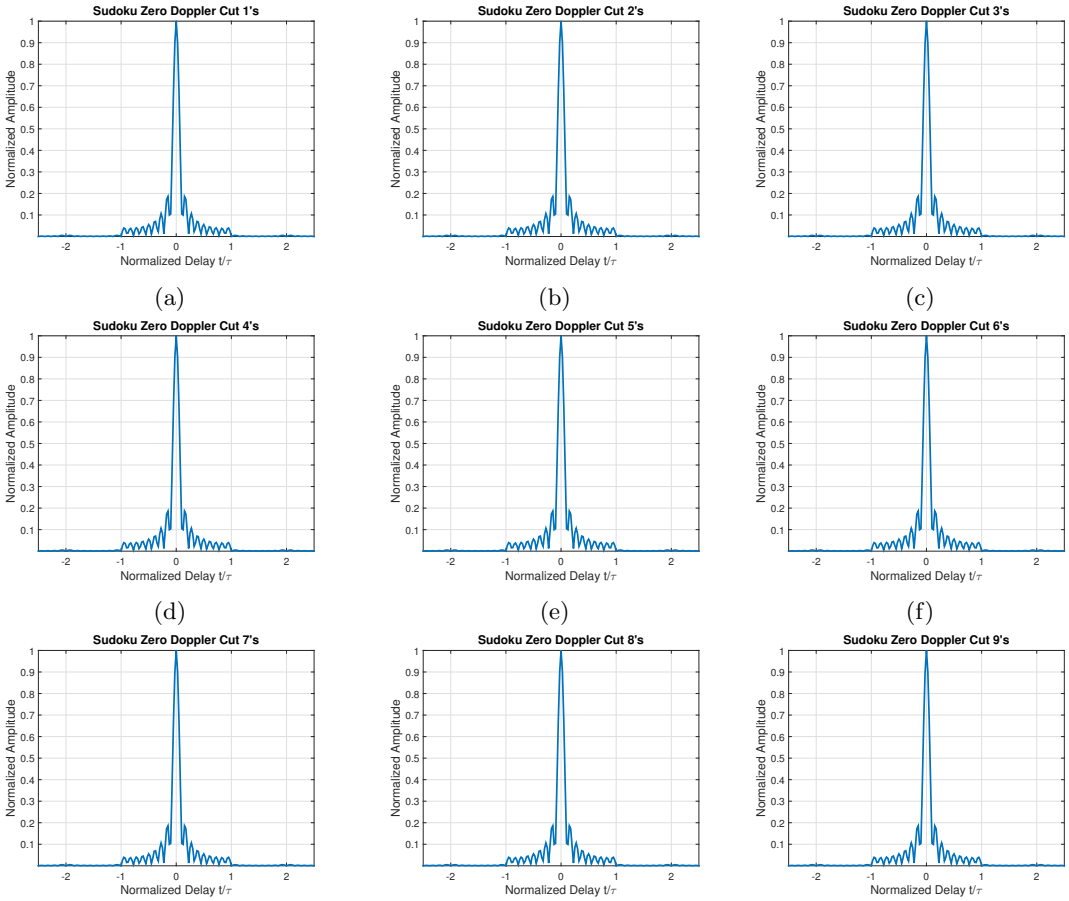
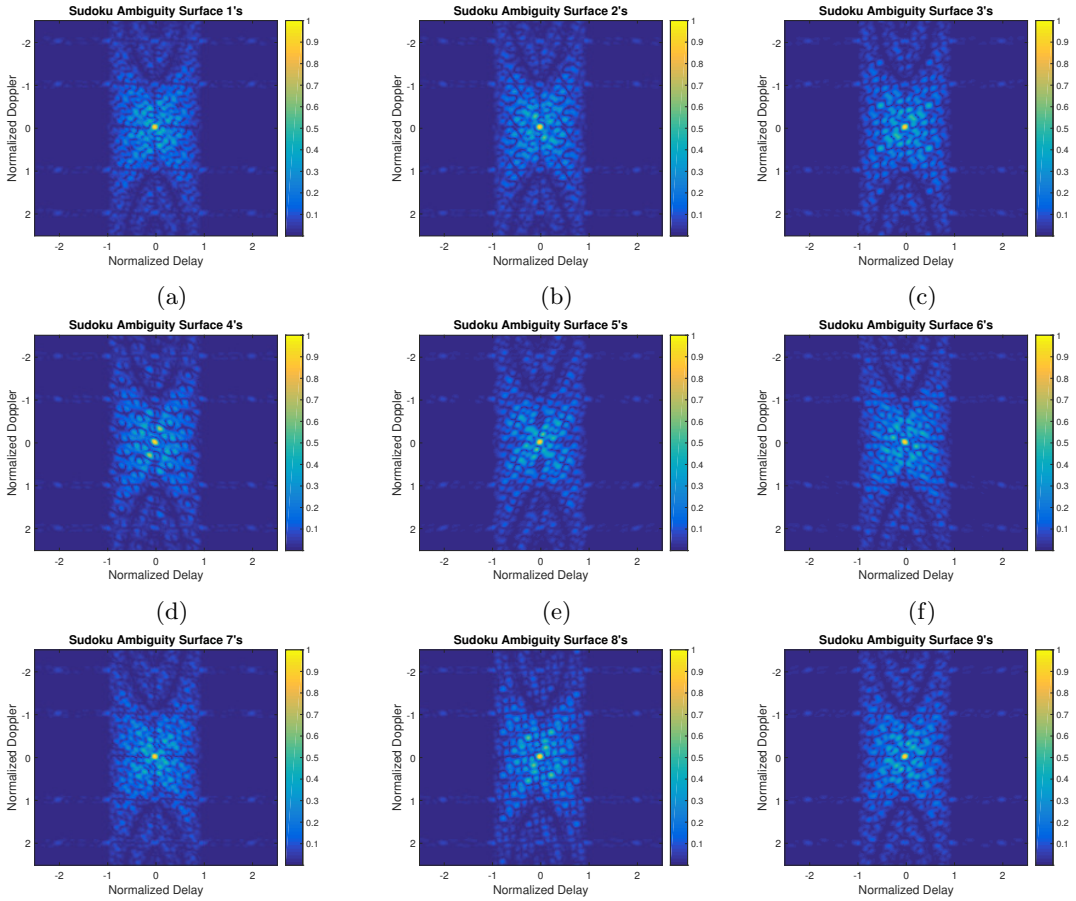


Figure 14: The zero Doppler cuts for the individual numbers of the previously described Sudoku puzzle solution. (a) number 1s, (b) number 2s, (c) number 3s (d) number 4s, (e) number 5s, (f) number 6s, (g) number 7s, (h) number 8s, (i) number 9s.

The auto-ambiguity surface for the individual Sudoku numbers in Fig. 11a are shown in Fig. 15. We see that for the most part all the numbers exhibit thumbtack like responses except for a couple of numbers which show strong secondary sidelobes such as in Fig. 15d. One of the questions we will further explore is which Sudoku codes make the best frequency coded sequences.



(a) (b) (c)

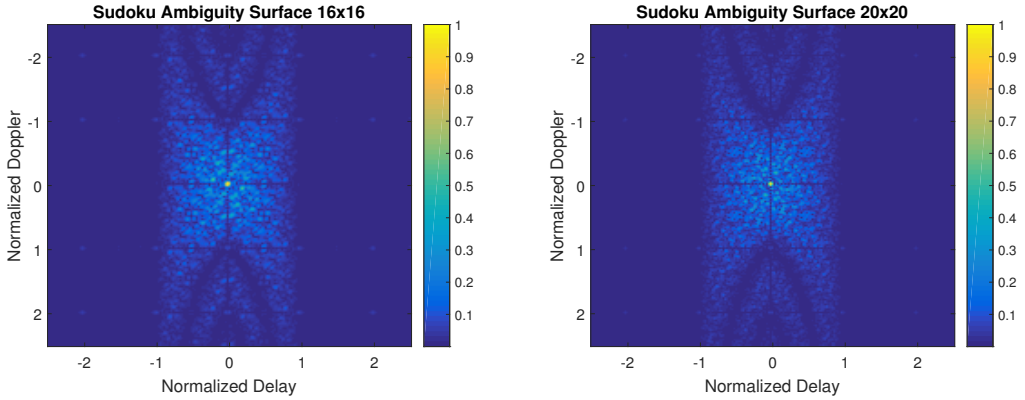
(d) (e) (f)

(g) (h) (i)

Figure 15: The auto-ambiguity surface of the individual numbers of the previously described Sudoku puzzle solution. (a) number 1s, (b) number 2s, (c) number 3s (d) number 4s, (e) number 5s, (f) number 6s, (g) number 7s, (h) number 8s, (i) number 9s.

2.2.1 Sudoku Size Variations

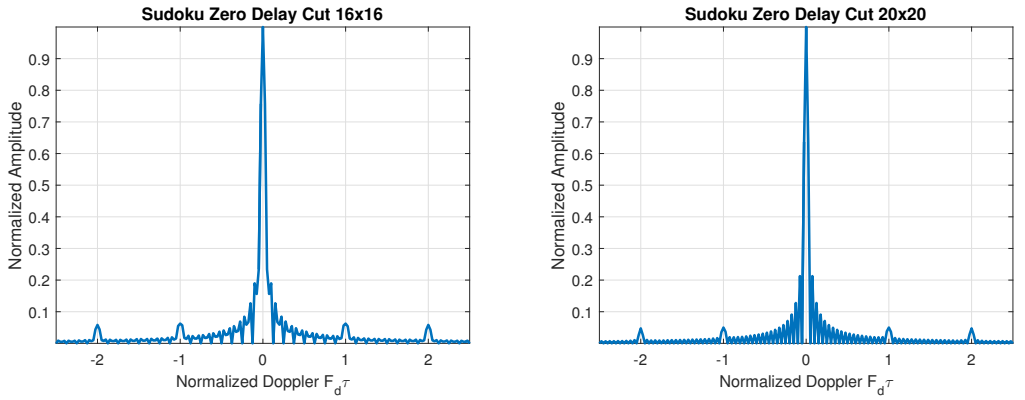
Sudoku puzzles can be found in various sizes besides the common 9×9. Large sizes include 16×16 and 20×20 as well as higher orders which often have unique names such as the 25×25 called "Sudoku the Giant". A comparison of larger Sudoku sequences are briefly examined in this section. The ambiguity surfaces in Fig. 16 show thumbtack like responses for the 16×16 and 20×20 Sudoku puzzles with low sidelobes compared to the main peak which becomes more narrow as the Sudoku sequence length is increased.



(a) (b)

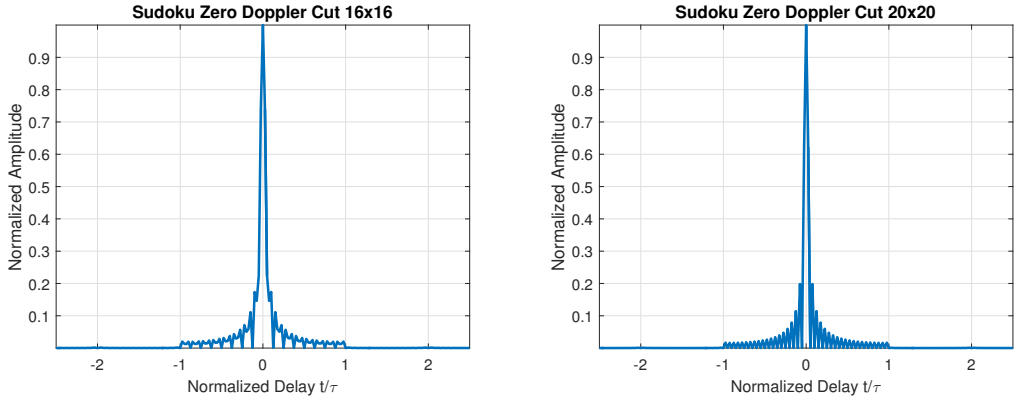
Figure 16: Larger Sudoku ambiguity surfaces. (a) 16×16 Sudoku puzzle solution. (b) 20×20 Sudoku puzzle solution.

The zero-delay cuts and zero-Doppler cuts for the 16×16 and 20×20 sized Sudoku solutions are shown in Fig. 17 and Fig. 18. We see as we increase the Sudoku sequence, the main peak becomes more sharply defined as well as the sidelobes become lower.



(a) (b)

Figure 17: Larger Sudoku zero-delay cuts. (a) 16×16 Sudoku puzzle solution. (b) 20×20 Sudoku puzzle solution.



(a) (b)

Figure 18: Larger Sudoku zero-Doppler cuts. (a) 16×16 Sudoku puzzle solution. (b) 20×20 Sudoku puzzle solution.

2.3 Sudoku Tiling

Due to the finite valid dimensions for a Sudoku puzzle, the idea of tiling is to concatenate smaller matrices to synthesize a larger matrix that could not normally be obtained. Fig. 19 shows a 9×9 Sudoku matrix with the corresponding locations of the 1s shown to the right of the matrix. Using the 1s as the Sudoku code we can link two together to create a 18×18 matrix shown in Fig. 20.

$$\left[\begin{array}{ccc|ccc|ccc} 1 & 7 & 2 & 5 & 4 & 9 & 6 & 8 & 3 \\ 6 & 4 & 5 & 8 & 7 & 3 & 2 & 1 & 9 \\ 3 & 8 & 9 & 2 & 6 & 1 & 7 & 4 & 5 \\ \hline 4 & 9 & 6 & 3 & 2 & 7 & 8 & 5 & 1 \\ 8 & 1 & 3 & 4 & 5 & 6 & 9 & 7 & 2 \\ 2 & 5 & 7 & 1 & 9 & 8 & 4 & 3 & 6 \\ \hline 9 & 6 & 4 & 7 & 1 & 5 & 3 & 2 & 8 \\ 7 & 3 & 1 & 6 & 8 & 2 & 5 & 9 & 4 \\ 5 & 2 & 8 & 9 & 3 & 4 & 1 & 6 & 7 \end{array}\right] \quad \left[\begin{array}{ccc|ccc|ccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{array}\right]$$

Figure 19: Sudoku solution of order 9, with the 1s pulled out to show one possible code.

$$\left[\begin{array}{ccc|ccc|ccc|ccc|ccc|ccc}
\mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0
\end{array}\right]$$

Figure 20: 18×18 array tiled array using two 9×9 Sudoku matrices.

The resulting ambiguity function surface is shown in Fig. 21a along with a normal 18×18 Sudoku puzzle in Fig. 21b. We can observe a focused peak in the ambiguity function surface of the tiled array in Fig. 21a and find it is comparable to the untiled array in Fig. 21b. Looking at the tiled case, more striations appear unlike the untiled sequence. We will see in the next section that as we further break down a matrix into smaller sub matrices, the resulting ambiguity function starts to resemble a linear up or down chirp, which may not come as a surprise as we are restricting the amount of deviation from a diagonal line by using smaller and smaller matrices.

The delay and Doppler cuts for the untiled and tiled 18×18 array is shown in Fig. 21. Both scenarios appear very similar to each other without much variation. The most noticeable differences are more easily seen in the 2D ambiguity surface plot.

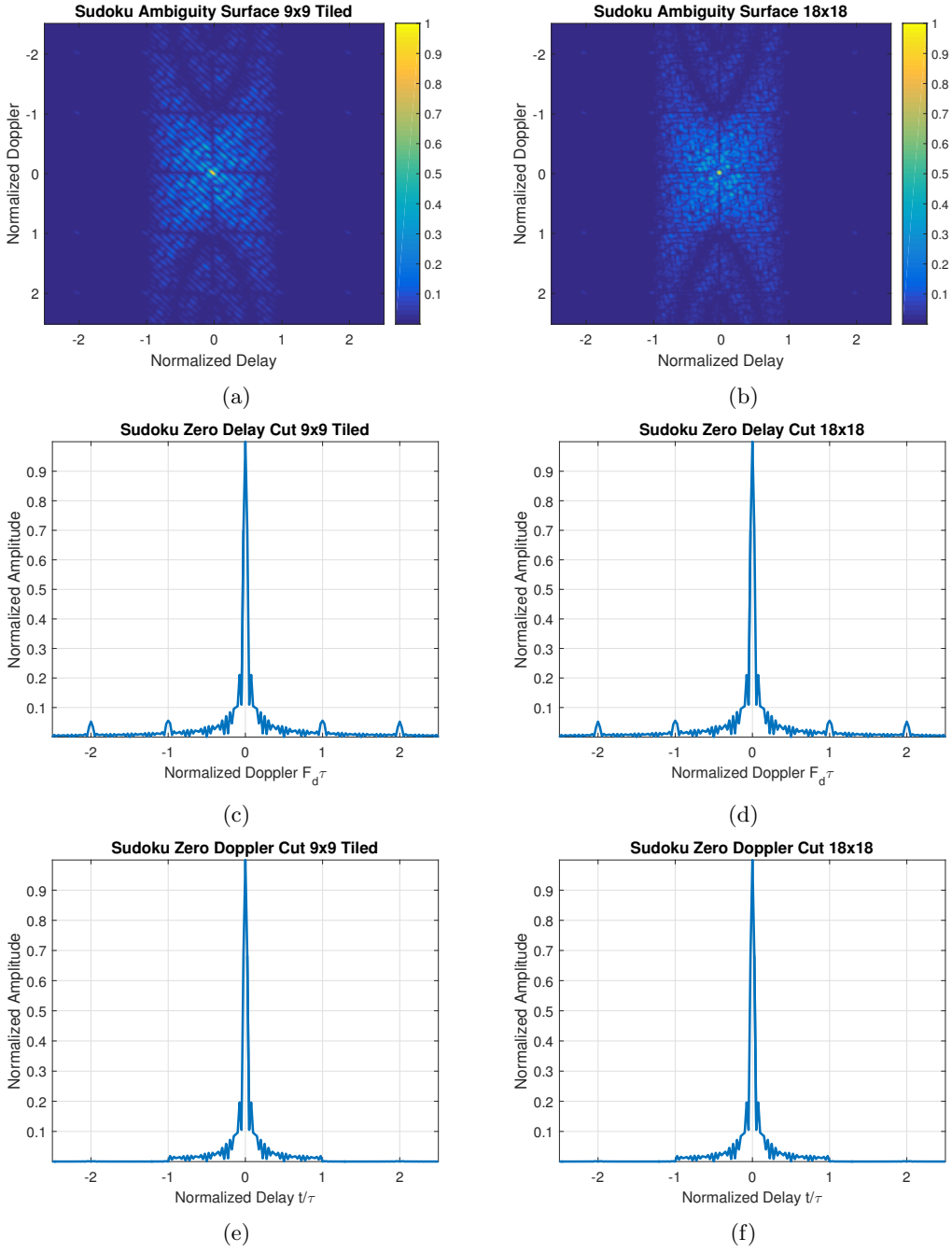
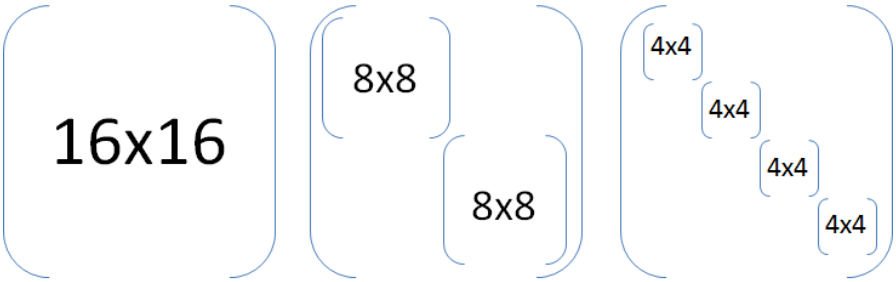


Figure 21: (a) Ambiguity surface for 18×18 tiled Sudoku array, (b) Ambiguity surface for 18×18 untiled Sudoku array, (c) Zero-delay cut for 18×18 tiled array, (d) Zero-delay cut for 18×18 untiled array, (e) Zero-Doppler cut for 18×18 tiled array, (f) Zero-Doppler cut for 18×18 untiled array.

2.3.1 Tile Comparison

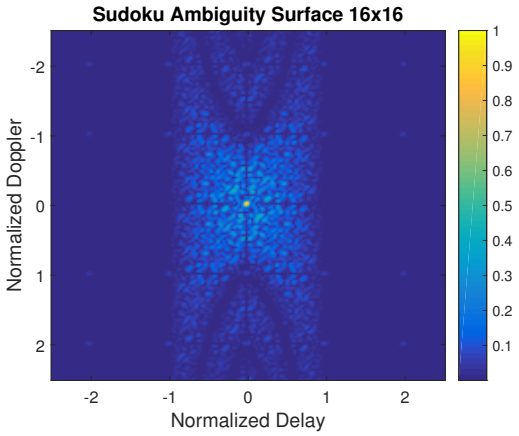
Further tile comparison is shown here where we start with a 16×16 Sudoku puzzle. The puzzle can be further broken down and tiled using two 8×8, four 4×4, and lastly eight 2×2.



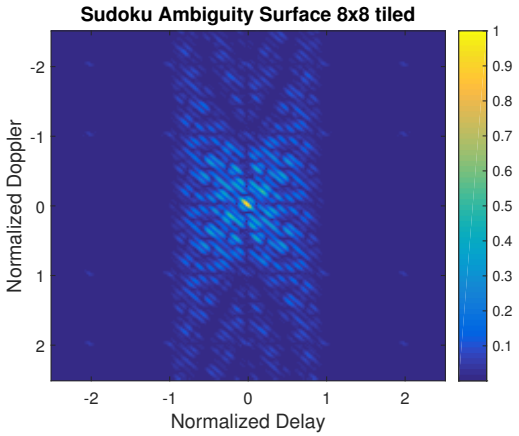
(a)

Figure 22: 16×16 Sudoku Puzzle tile comparison.

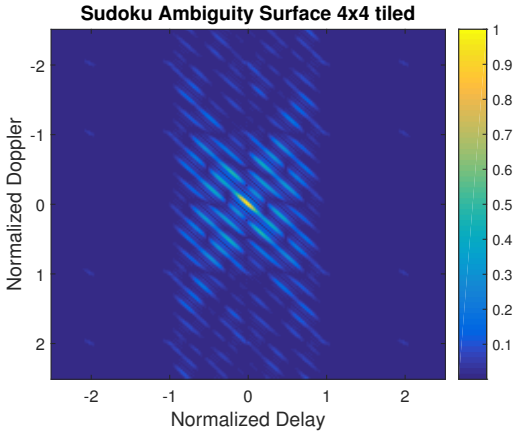
The different tiling results are shown in Fig. 23 where we see that in Fig. 23a, the response is a large peak in the center of the image. Breaking the matrix into smaller sub matrices then, we see start to see that the resulting ambiguity function resembles that of the chirp waveform as evident in Fig. 23b and Fig. 23c. This should be clear as we are utilizing smaller and smaller matrices the randomness becomes limited and deviates less from a straight upchirp or downchirp.



(a)



(b)



(c)

Figure 23: Ambiguity surface plots for different number of Sudoku tiling matrices of a 16×16 Sudoku matrix. (a) Untiled 16×16 ambiguity plot (b) two 8×8 matrix tiling, (c) four 4×4 matrix tiling.

2.4 Waveform Doppler Tolerance

One characteristic that may be of interest is how Doppler tolerant are Sudoku frequency hopped waveforms. First, let us return to the simple rectangular pulse and LFM waveforms. The rectangular pulse at different Doppler cuts can be seen in Fig. 24b for ambiguity Doppler shift cuts of 0.2τ, 0.6τ, and 2τ, We see that as the Doppler shift increases, the main peak decreases until it becomes noise like at just a couple multiples of $1/\tau$.

Looking at the LFM waveform Doppler tolerance in Fig. 24c, we observe that due to the range-Doppler skew, a Doppler shift will result in a error in the range of a target; however this is not such a bad feature as the high ridge of the LFM waveform will still result in a large matched filter response from the target, just not at the correct delay. Costas codes and Sudoku codes provide thumbtack like responses which exhibit low Doppler tolerance. A Doppler shift for a Costas code will cause the correlation response to shift from the center giving a low return as shown in Fig. 24a. Therefore, based on the Costas results, we can conclude that Sudoku based sequences are not Doppler tolerant.

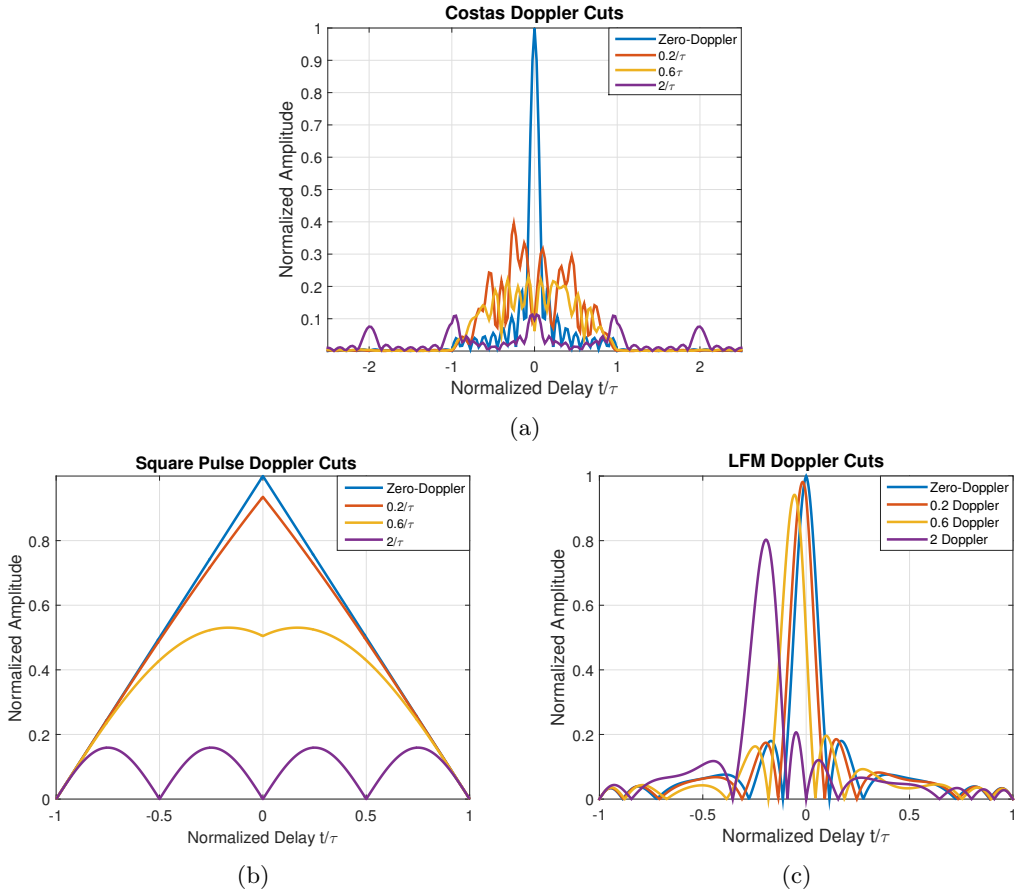


Figure 24: Doppler tolerances of different types of waveforms. (a) Costas coded waveform, (b) Rectangular pulse, (c) LFM waveform.

2.5 Sudoku Vs Costas Coded Ambiguity Functions

Instead of evaluating equation (5) for side-lobes of multiple sequences, which would be a time consuming process, we can instead analyze the auto coincidence and cross coincidence matrices which can be performed more efficiently through 2D correlation. We examine Sudoku puzzles as applications for frequency coded waveforms in more detail, including statistical probability of different codes producing different number of coincidences, good and bad Sudoku codes, lastly we want to answer how Sudoku codes compare to Latin Squares and if the added box constraints adds any advantages.

2.5.1 Sudoku Backtracing Algorithm

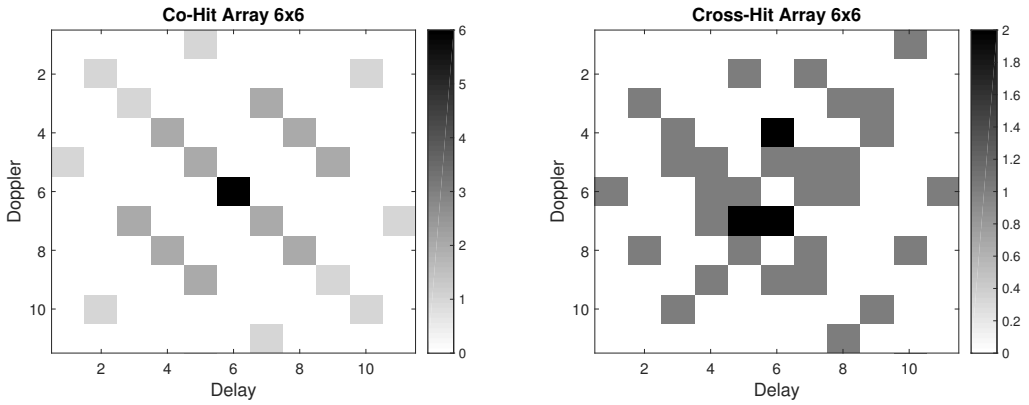
In order to gather statistics on multiple Sudoku puzzles, it was required to look into the generation of numerous puzzles to examine. The generation of the coincidence matrices was accomplished through the backtracking algorithm for both the Sudoku and the Latin Square matrices. The backtracking algorithm recursively solves a given Sudoku or Latin Square grid. We start with a blank grid beginning in the upper left corner, continue column wise until we reach the end, and then proceed onto the next row. The solver inserts a valid number for a given square from a uniform distribution while checking to make sure it satisfies box, row, and column constraints. The algorithm will backtrack to previous squares if no valid numbers solve the current square.

Increasing the Sudoku and Latin Square order also increased the number of possible puzzles. While we could easily sort through all 4×4 order matrices, going through 9×9 and even high orders is not feasible. Therefore, we generated a finite number of puzzles and evaluated them for different criteria. Going above a 12×12 Sudoku puzzle was not practical as the computational time for the backtracking algorithm increases significantly for a valid solution; thus 12×12 was the highest order investigated. The number of puzzles generated was 10,000 for a given size whose auto and cross coincidence results are examined in the next section.

2.6 Sudoku Puzzle Coincidence Analysis

The auto coincidence matrix or co-hit array is computed by correlating the binary frequency matrix with itself, while the cross coincidence or cross-hit array is computed by correlating two different frequency codes with each other. By examining the resulting hit array matrices and sorting the resulting collisions in order from greatest to least, we can investigate how well these codes perform. Finding the second highest number of collisions in the co-hit array will give the expected sidelobe performance of that particular code, while the highest number of collisions given in the cross-hit array will give the highest side-lobe coincidence for interference. An example of the coincidence arrays is given in Fig. 25.

After the generation of every Sudoku puzzle that particular puzzle is examined by taking each of the N codes and performing their co-hit array analysis. Similarly, the cross-hit



(a) (b)

Figure 25: Example co-hit and cross-hit arrays. (a) Co-Hit array for the sequence [2,5,4,3,6,1] (b) Cross-Hit array for the sequences [2,5,4,3,6,1] and [1,3,5,6,4,2].

array analysis is performed by taking every combination of the N codes and evaluating them against eachother. This analysis is performed for Sudoku and Latin Square matrices to compare their results.

2.6.1 Co-hit Array Analysis

Tables 3 and 4 show the percentages of the co-hit array analysis for the different size Sudoku and Latin Square matrices. We can see from the tables that the Sudoku co-hit array analysis that has the arrays increase in size the average collision level shifts to the right with the 6×6 averaging 2 collisions, and 12×12 averaging 3 for the majority of collisions. The Latin Squares perform similar to the Sudoku matrices with slightly higher averages of collisions which will be more evident in the histograms.

Table 3: Percentage of Auto Coincidences based upon Sudoku grid size.

Sudoku Grid	1	2	3	4	5	6
4×4	50%	50%	0%	0%	0%	0%
6×6	22.21%	55.8%	19.93%	2.04%	0%	0%
9×9	0.20%	50.24%	42.02%	6.56%	0.88%	0%
12×12	0%	24.51%	58.44%	15.16%	1.97%	0.22%

Histograms displaying the number of auto-coincidences are tabulated for 6×6, 9×9, and 12×12 Sudoku and Latin Square matrices, these plots and percentages are shown in Fig. 26, Fig. 27 and Fig. 28. Examining the results show that the majority of Sudoku puzzles have 2 or 3 coincidences with 4×4 containing equal amounts of 1 and 2

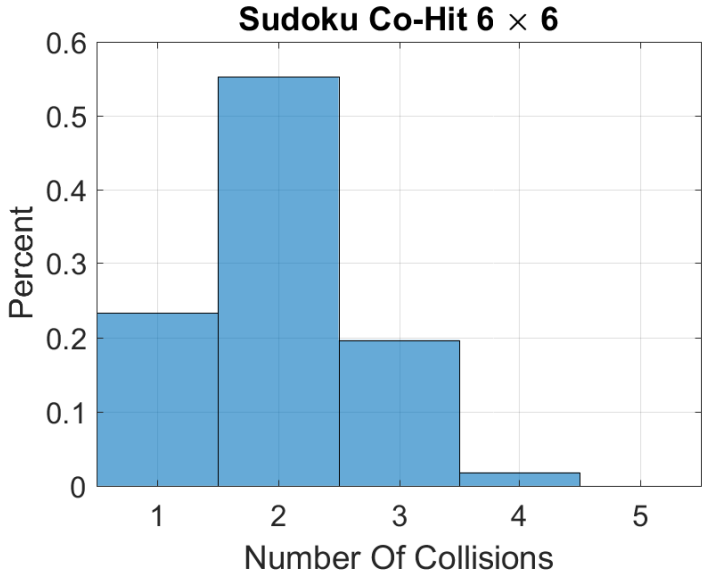
Table 4: Percentage of Auto Coincidences based upon Latin Square grid size.

Latin Square	1	2	3	4	5	6	7
4×4	50%	50%	0%	0%	0%	0%	0%
6×6	15.15%	59.5%	21.93%	2.88%	0.50%	0%	0%
9×9	0.20%	45.54%	44.15%	8.75%	1.15%	0.17%	0.01%
12×12	0.0025%	22.67%	58.34%	16.24%	2.54%	0.30%	0.033%

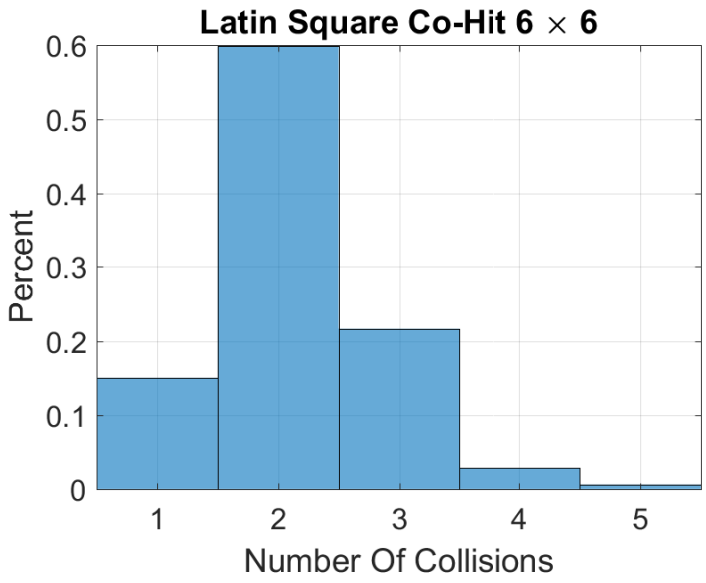
coincidences and becoming more spread as the Sudoku order is increased, where we still see that the majority of coincidences are still 2 for all orders investigated.

Comparing the Sudoku co-hit arrays with Latin Squares, we find that the Sudoku puzzles offer slightly better sidelobe performance with a greater percentage of Sudoku frequency codes offering "1" and "2" number of coincidences. The numerical simulations and analysis show that Sudoku frequency codes offer slightly better performance.

Furthermore, we can examine the cumulative coincidences for a given Sudoku or Latin Square matrix. The cumulative total is defined as taking a generated matrix and calculating the number of collisions for each of the N codes and summing the total together. Histograms comparing the Sudoku and Latin Squares cumulative distributions are shown in Fig. 29, where we see that the Sudoku results are slightly shifted to the left, indicating better performance for a particular puzzle order.

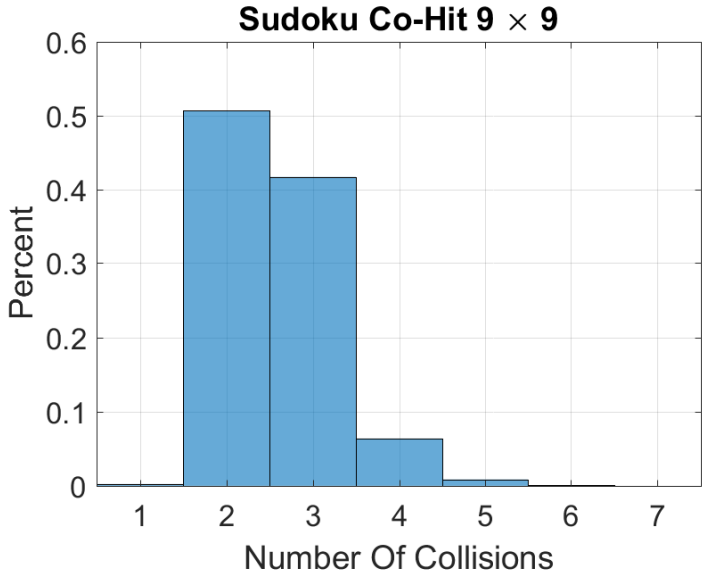


(a)

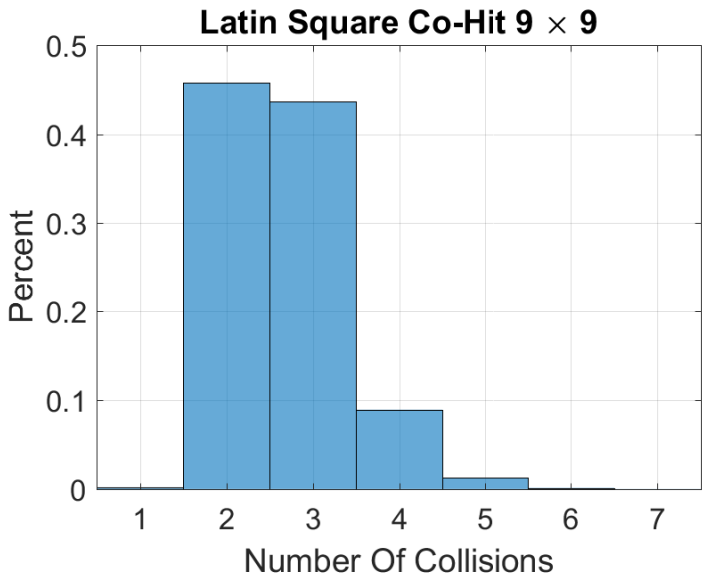


(b)

Figure 26: Histogram co-hit array Collisions. (a) 6×6 Sudoku grid, (b) 6×6 Latin Square.

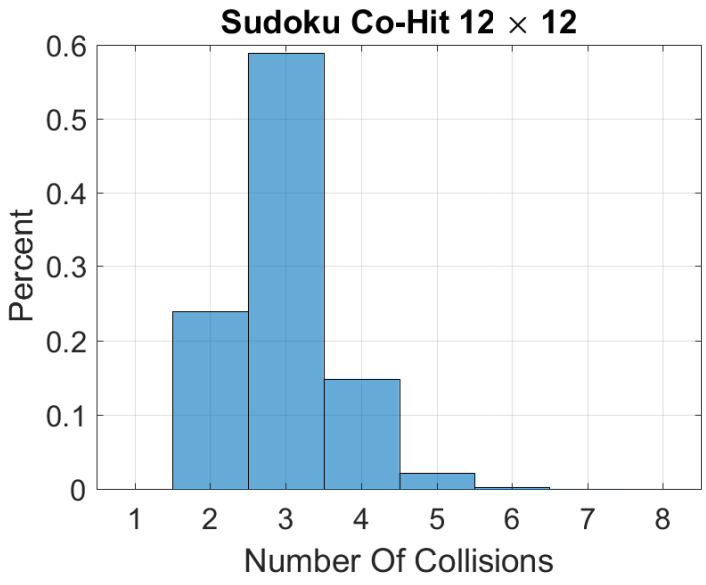


(a)

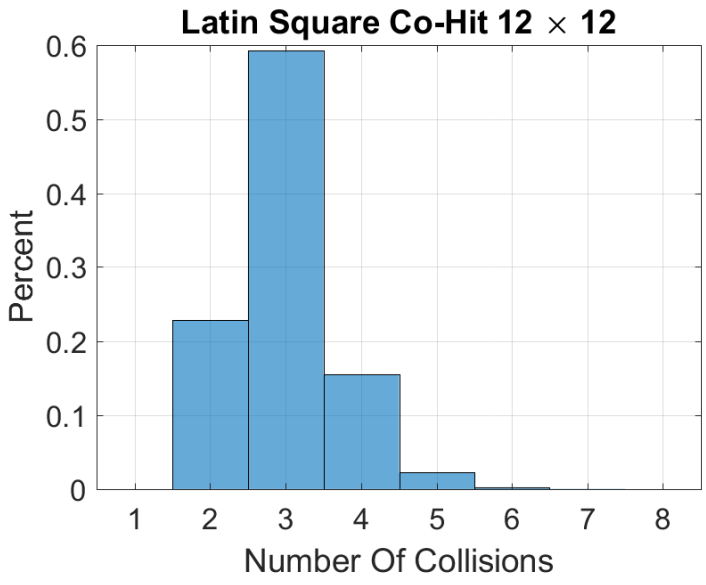


(b)

Figure 27: Histogram co-hit array Collisions. (a) 9×9 Sudoku grid, (b) 9×9 Latin Square.

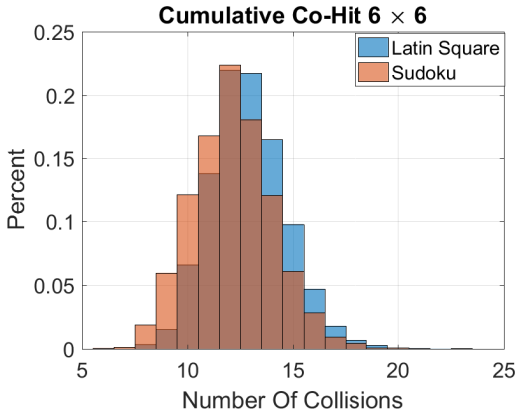


(a)

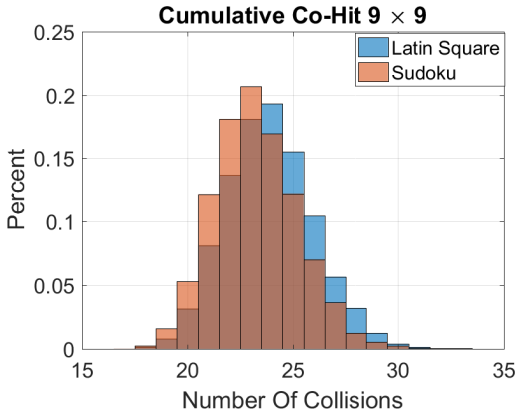


(b)

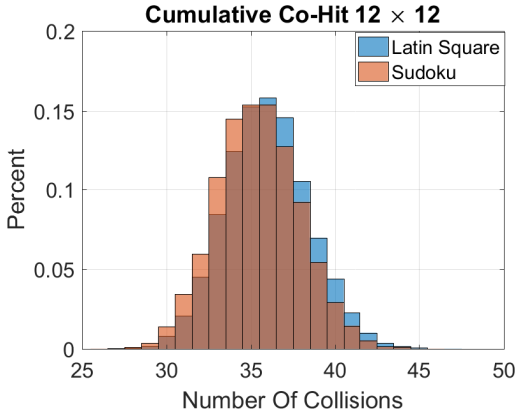
Figure 28: Histogram co-hit array Collisions. (a) 12×12 Sudoku grid, (b) 12×12 Latin Square.



(a)



(b)



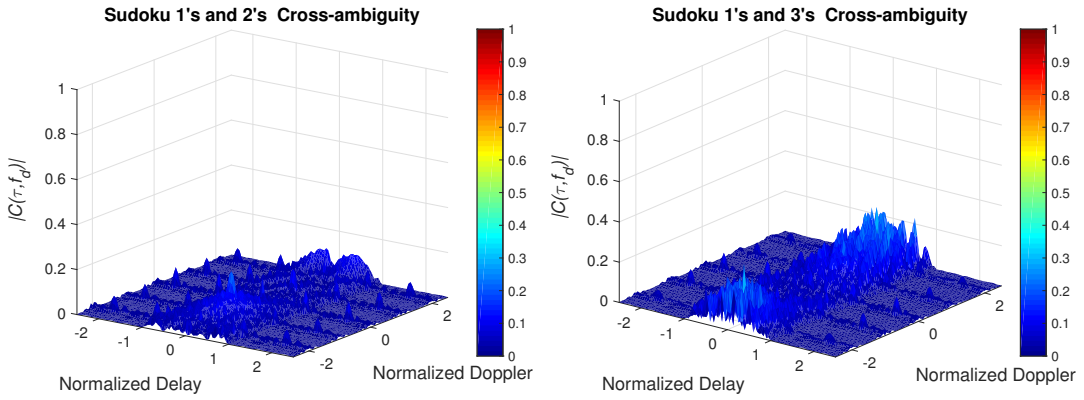
(c)

Figure 29: Sudoku and Latin Square cumulative co-hit collisions. (a) 6×6 cumulative co-hit collisions, (b) 9×9 cumulative co-hit collisions, (c) 12×12 cumulative co-hit collisions.

2.6.2 Cross-hit Array Analysis

The cross-ambiguity function previously mentioned is the matched filter output from two different signals. One reason for exploring cross-ambiguity is see how interference from other waveforms affects the matched filter output. Here we examine Sudoku solutions as a means for N users to communicate so for a 9×9 Sudoku matrix the number of users would be $N = 9$. We compare Latin Square permutations to that of the Sudoku solutions by means of the backtracking algorithm to generate 10,000 different matrices for analysis.

The results in Fig. 30 show the cross ambiguity results for the Sudoku matrix in Fig. 11a where we used the integers 1,2 and 3 to compute the cross ambiguity results. We see for the most part there exists a low number of collisions and the sidelobes are away from the zero Doppler and delay intersection.



(a) (b)

Figure 30: Cross ambiguity for a Sudoku puzzle. (a) 1s and 2s cross ambiguity, (b) 1s and 3s cross ambiguity.

Evaluating numerous Sudoku puzzles for their cross ambiguity properties is not feasible by computing the actual cross ambiguity function. Instead, we use the cross hit array matrix to evaluate the interference among the frequency codes. Figs. 31-33 and Tables 5-6 show the spread of the cross-hits among different size Sudoku and Latin Square permutations.

Tables 5-6 show as the Sudoku order increases, the majority of cross-hits shifts to the right and the same can be seen for the Latin Square puzzles. The Latin Squares show a slightly smaller shift than the Sudoku matrices and therefore perform marginally better for cross-hit performance.

We show results for two different scenarios. First, for a given Sudoku or Latin Square Matrix, we evaluate the matrix as a whole, that is, we compute the average cross hit sidelobes for the N frequency codes. Secondly, we evaluate the cumulative cross-hit arrays for each of the Sudoku and Latin Squares produced.

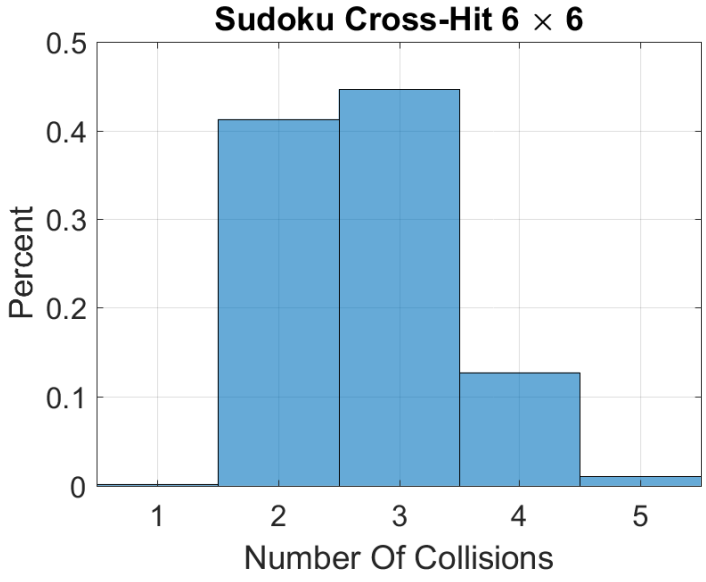
Table 5: Percentage of Cross Coincidences based upon Sudoku grid size.

Sudoku Grid	1	2	3	4	5	6	7
6×6	0.20%	41.31%	44.6%	12.7%	1.08%	0%	0%
9×9	0%	10.4%	63.2%	22.2%	3.5%	0.5%	0%
12×12	0%	1.66%	56.28%	34.96%	6.15%	0.8%	0.1%

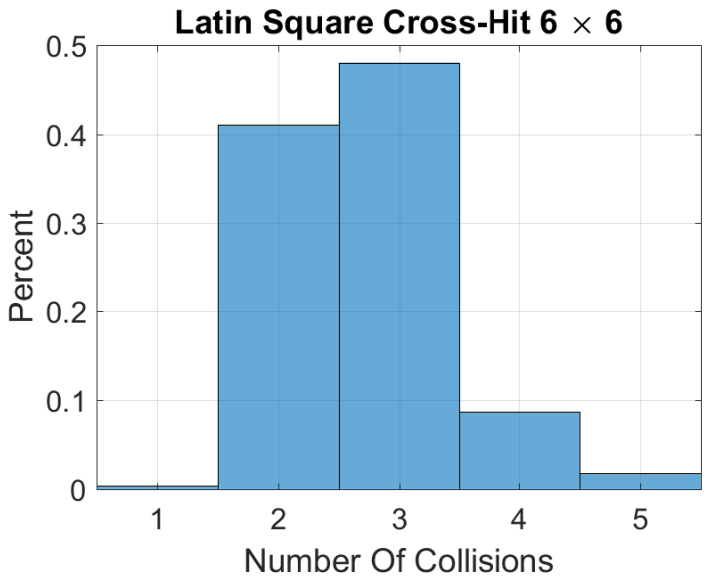
Table 6: Percentage of Cross Coincidences based upon Latin Square grid size.

Latin Square	1	2	3	4	5	6	7	8
6×6	0.36%	41.1%	48.6%	8.69%	1.79%	0%	0%	0%
9×9	0%	11.95%	64.6%	20.2%	2.9%	0.31%	0.02%	0%
12×12	0%	1.97%	59.00%	33.07%	5.24%	0.6%	0.06%	0%

Histograms displaying the number of cross-hit coincidences for 6×6, 9×9, and 12×12 Sudoku and Latin Square matrices are shown in Figs. 31-33. Viewing these results in graph form rather than tabular form clearly shows the trend of shifting to the right as the order of the Sudoku and Latin Square matrices increases. The cumulative cross-hit array analysis in Fig. 34 shows that indeed the Latin Squares offer slightly better performance in cross-hit performance. Comparing these results to the co-hit array results, we see a trade-off between co-hit and cross-hit performance of Latin Squares and Sudoku based matrices.

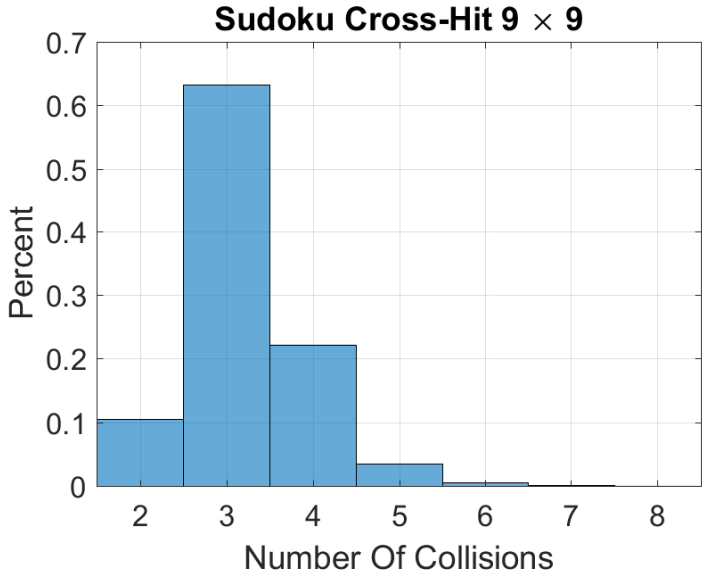


(a)

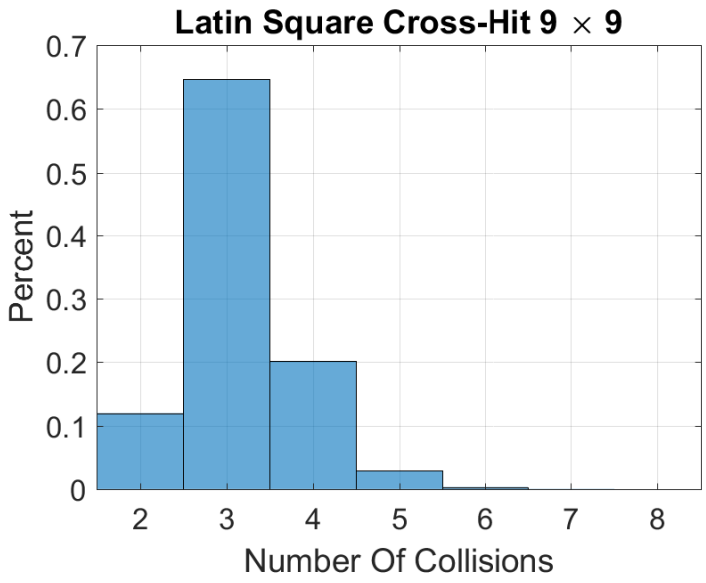


(b)

Figure 31: Latin Square and Sudoku individual cross-hit collisions. (a) 6×6 Sudoku cross-hit collisions, (b) 6×6 Latin Square cross-hit collisions.

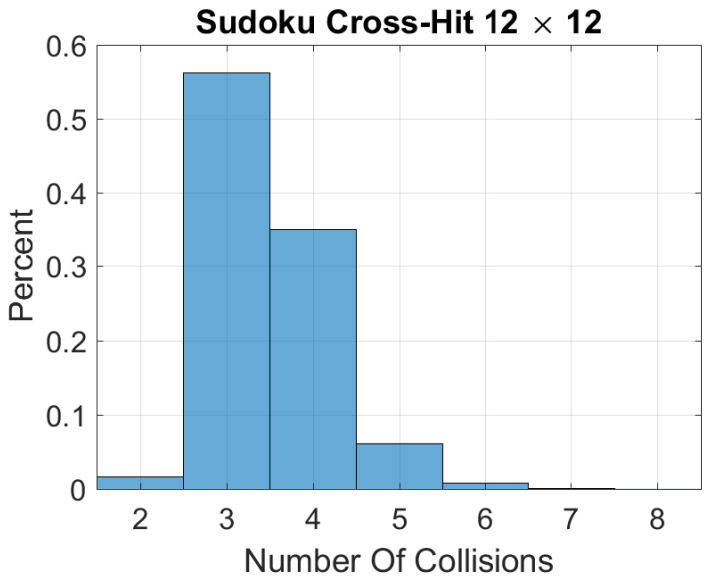


(a)

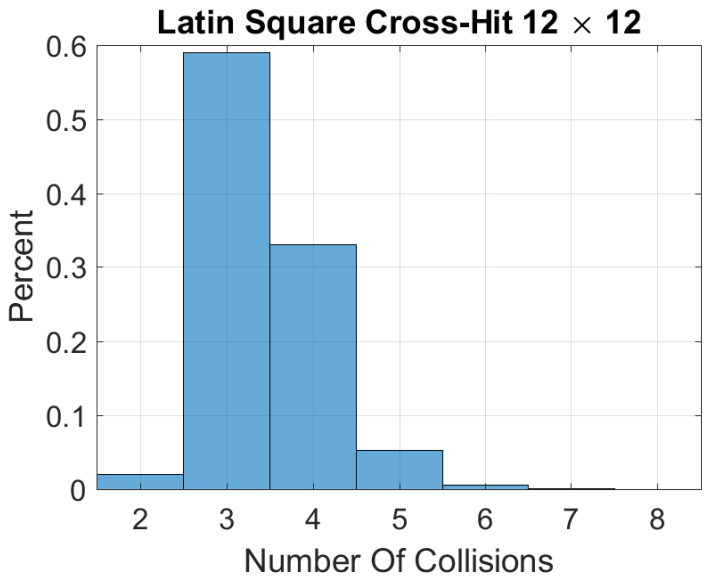


(b)

Figure 32: Latin Square and Sudoku individual cross-hit collisions. (a) 9×9 Sudoku cross-hit collisions, (b) 9×9 Latin Square cross-hit collisions.

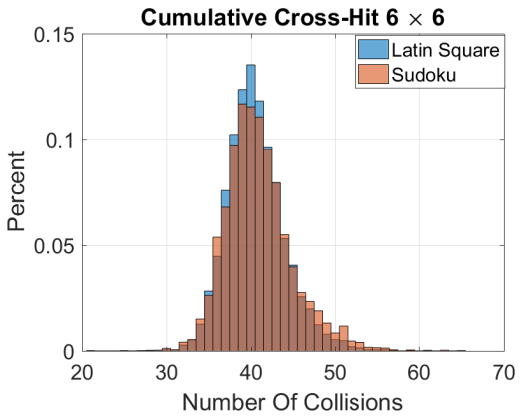


(a)

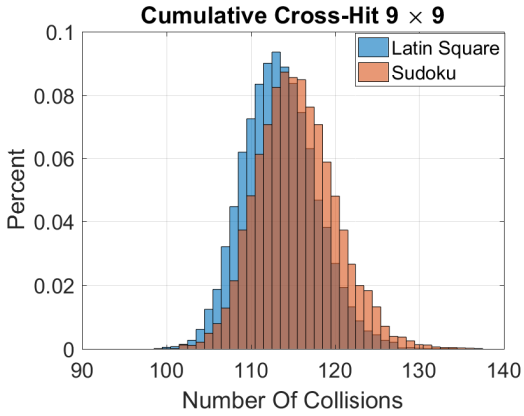


(b)

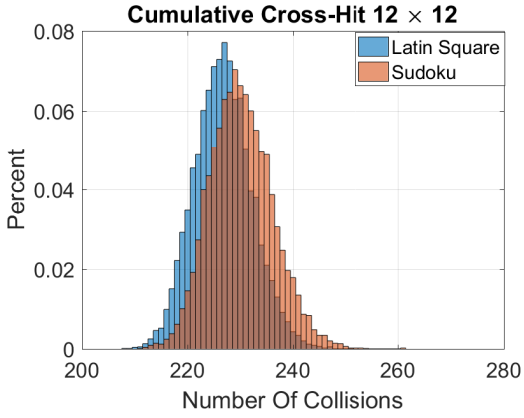
Figure 33: Latin Square and Sudoku individual cross-hit collisions. (a) 12×12 Sudoku cross-hit collisions, (b) 12×12 Latin Square cross-hit collisions.



(a)



(b)



(c)

Figure 34: Latin Square and Sudoku cumulative cross collisions. (a) 6×6 cumulative cross collisions, (b) 9×9 cumulative cross collisions, (c) 12×12 cumulative cross collisions.

2.7 Costas Sudoku Solutions

Within the Sudoku context, their exist a subset of Sudoku codes that have Costas properties. Starting the search with 4×4 Sudoku matrices, we came across the following solution depicted in Fig. 35. Regardless of which number locations is observed, they are in fact identical with different rotations applied. For example, flipping the 2s vertically causes the 2s to line up with the 4s. Fig. 35 shows a 4×4 Sudoku matrix in which every subset of the same number is a Costas sequence, meaning N Costas sequences can form a valid Sudoku puzzle and fit on the same $N \times N$ matrix. Therefore, some Sudoku solutions are in fact Costas arrays. However, not all Costas arrays are Sudoku solutions, as shown in Fig. 36, which demonstrates a violation of the Sudoku box constraint.

$$\left[\begin{array}{cc|cc} 1 & 2 & 3 & 4 \\ 4 & 3 & 2 & 1 \\ \hline 2 & 1 & 4 & 3 \\ 3 & 4 & 1 & 2 \end{array}\right]$$

Figure 35: Sudoku 4×4 search result where every number subset results in a Costas sequence.

$$\left[\begin{array}{cc|cc} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ \hline 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{array}\right]$$

Figure 36: Costas code that is not a valid Sudoku solution.

Another example was found when looking at 9×9 Sudoku matrices. The sequence $[9, 1, 4, 5, 3, 7, 6, 8, 2]$ corresponds to the locations of the 1s in the following Sudoku matrix. This sequence is a Costas code but cannot be translated or rotated to fill in the remaining spaces.

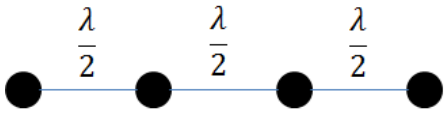
$$\left[\begin{array}{ccc|ccc|ccc} 1 & 4 & 7 & 8 & 2 & 9 & 5 & 3 & 6 \\ 2 & 6 & 9 & 5 & 4 & 3 & 8 & 1 & 7 \\ 5 & 3 & 8 & 6 & 7 & 1 & 9 & 2 & 4 \\ \hline 3 & 8 & 2 & 7 & 6 & 4 & 1 & 9 & 5 \\ 7 & 9 & 6 & 1 & 3 & 5 & 4 & 8 & 2 \\ 4 & 5 & 1 & 9 & 8 & 2 & 6 & 7 & 3 \\ \hline 6 & 7 & 3 & 4 & 1 & 8 & 2 & 5 & 9 \\ 8 & 2 & 5 & 3 & 9 & 6 & 7 & 4 & 1 \\ 9 & 1 & 4 & 2 & 5 & 7 & 3 & 6 & 8 \end{array}\right]$$

Figure 37: 9×9 Sudoku puzzle in which the 1's correspond to a valid Costas code.

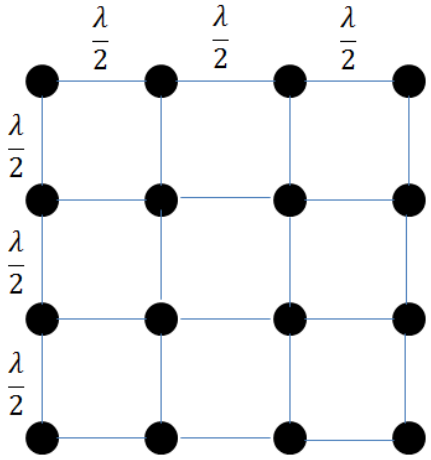
3 Phased Antenna Arrays

Antenna arrays are a collection of usually similar elements arranged in a geometric lattice. Antenna arrays have received a large amount of research due to their ability to change the antenna pattern through variables of element spacing and excitation amplitude. The term phased array comes from the fact that by altering the phases of the individual elements, we can steer the beam in the azimuth or elevation directions depending upon one or two dimensional arrays. Compared to mechanical steered arrays, phased arrays can steer the beam significantly quicker.

Antenna arrays can have many different lattice structures. Some of the more common geometric examples are linear lattices shown in Fig. 38a, where the elements are arranged linearly and planar designs where the elements are arranged in two-dimensional patters such as square and rectangular patterns, shown in Fig. 38b.



(a) Linear array lattice structure.



(b) 2D planar array lattice structure.

Figure 38: Example geometric phased antenna lattice structures.

The spacing of the periodic antenna array elements design dictates if grating lobes will appear. Deciding upon the element spacing depends upon whether we are operating on broadside (90°) or endfire (0° or 180°). The spacing requirement for linear arrays is given in equation (8) where λ is the operating wavelength, and θ_0 is the main beam pointing direction:

$$d < \frac{\lambda}{1 + |\sin\theta_0|} \ . \tag{8}$$

Having an equal element spacing greater than $\lambda/2$ will cause grating lobes to appear. An alternative method to prevent grating lobes is having irregular spacing in the geometry of the array. Utilizing Sudoku matrices, we examine their potential for the synthesis of randomly spaced arrays.

Let us first familiarize ourselves with the linear array. The linear array factor AF is given in equation (9), where A_n is the excitation amplitude, N is the number of elements, and $\psi = \beta d \cos\theta + \alpha$, where β is the wave number, d is the distance, θ is the look angle

$$AF = \sum_{n=0}^{N-1} A_n e^{jn\psi}. \tag{9}$$

A maximum occurs at $\psi = 0$ which means $\alpha = -\beta d \cos\theta_0$, where θ_0 is the direction in which we want the array factor to be at maximum. Therefore, we can scan the array to a desired angle by modifying equation (9) so that it incorporates the desired phase shift at each element to achieve the desired direction of scan as folllows

$$AF = \sum_{n=0}^{N-1} A_n e^{jn(\beta d \cos\theta - \beta d \cos\theta_0)}. \tag{10}$$

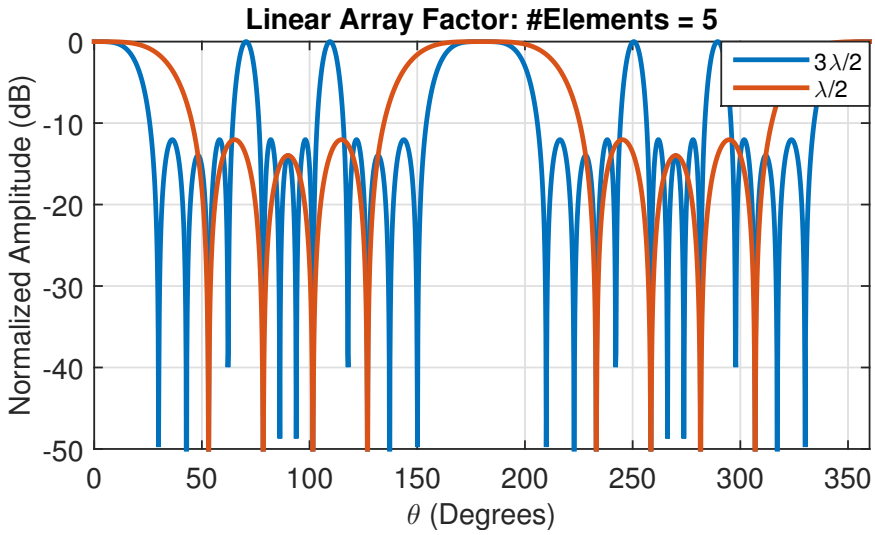
A finite number of variables are usually modified when dealing with array synthesis, which include element spacing, excitation amplitude, and array thinning. When element spacing is equal in both planar directions, we have a periodic antenna spacing. An example of the appearance of grating lobes is shown in Fig. 39 where once the spacing is greater than $\lambda/2$, we see the appearance of grating lobes for periodic spacing.

3.1 Periodic Planar Arrays

Looking at geometric arrangements in multiple dimensions, the planar lattice structure is quite common with arrangements in rectangular, triangular, and hexagonal patterns. We only consider the rectangular arrangement with the planar array factor [21] given as

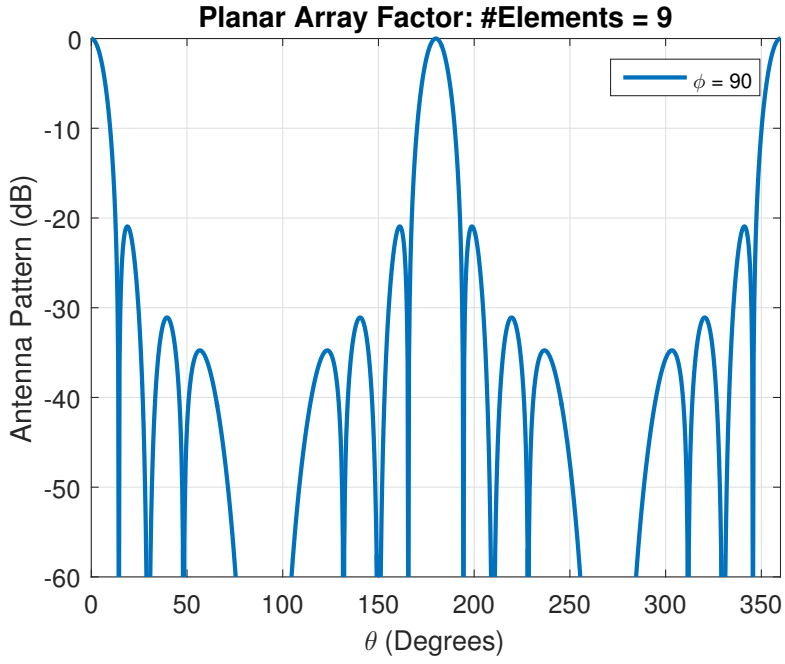
$$AF(\theta,\phi) = \sum_{n=1}^{N}\sum_{m=1}^{M} I_{mn} \mathrm{e}^{j\beta[d_x(m-1)(\sin\theta\cos\phi - \sin\theta_0\cos\phi_0) + d_y(n-1)(\sin\theta\sin\phi - \sin\theta_0\sin\phi_0)]}. \tag{11}$$

where β is the wave number, I_{mn} is the excitation current for the element at position (m,n), θ and ϕ are the elevation and azimuth angles, θ_0 and ϕ_0 are the main beam pointing direction, and d_x and d_y is the element spacing in wavelengths in the x and y directions, respectively. Similar to the linear array arrangement, the spacing in the x and y directions affect the appearance of grating lobes and should generally be less than $\lambda/2$.



(a)

Figure 39: Linear array with 5 elements demonstrating the appearance of grating lobes when the element spacing is too large.

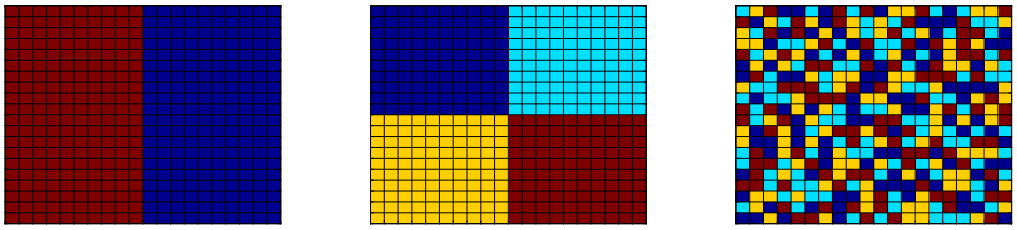


(a)

Figure 40: Planar array with 9 elements and spacing of $\lambda/2$ in the $\phi = 90°$ cut.

3.2 Sudoku Interleaved Arrays

This section explores two applications of Sudoku puzzles to antenna arrays through interleaving of the elements: (1) main beam steering in multiple directions using a fraction of the array elements, and (2) generation of simultaneous multiple beams. Array interleaving has been studied previously [22] and is of interest when multiple arrays have to share the same area. Multiple arrangements can be made to have arrays which operate at the same frequency as shown in Fig. 41. Fig. 41a shows two arrays positioned side-by-side, Fig. 41b shows an arrangement for four arrays, and Fig. 41c shows a Sudoku based interleaving for four arrays. The non-interleaved arrays have better sidelobes but the interleaved arrays offer narrower beamwidths [22].



(a) (b) (c)

Figure 41: Planar array layout for multiple arrays sharing the same area: (a) Side-by-side layout; (b) Quadrant layout for four separate arrays; (c) Sudoku based layout for four arrays.

We consider a 20×20 planar array which consists of 400 elements. The 20×20 array will have four different beams where the Sudoku puzzle is used to separate the elements into four groups based upon their numbers as shown in Table 7. In other words, 25% of the elements will be used to steer the beam in a desired direction. A comparison of the Sudoku interleaving to that of the quadrant layout in Fig. 41b for the XZ plane ($\phi = 0°$) is shown in Fig. 42, where we can see the narrower beamwidth of the interleaved array.

Table 7: Planar array steering phase groups.

Group Number	1	2	3	4
Numbers	1,2,3,4,5	6,7,8,9,10	11,12,13,14,15	16,17,18,19,20
Scan Angle	0°	20°	40°	60°

First, we only consider a single main beam which is steered to 40° in the θ direction as well as a single beam steered to 40° in both the θ and ϕ directions. The ideal case in which all elements are turned on is shown in Fig. 43.

We next compare the ideal scenarios using two approaches: the first utilizing Sudoku puzzles while the second being formed by a random permutation of the array elements.

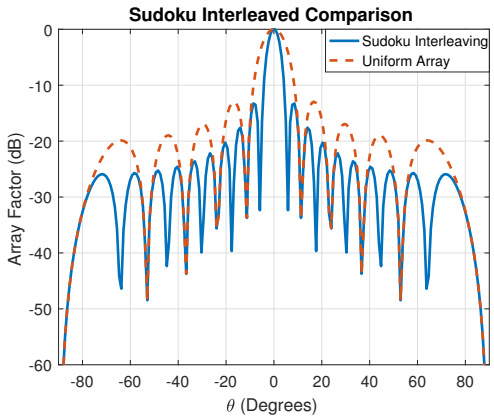
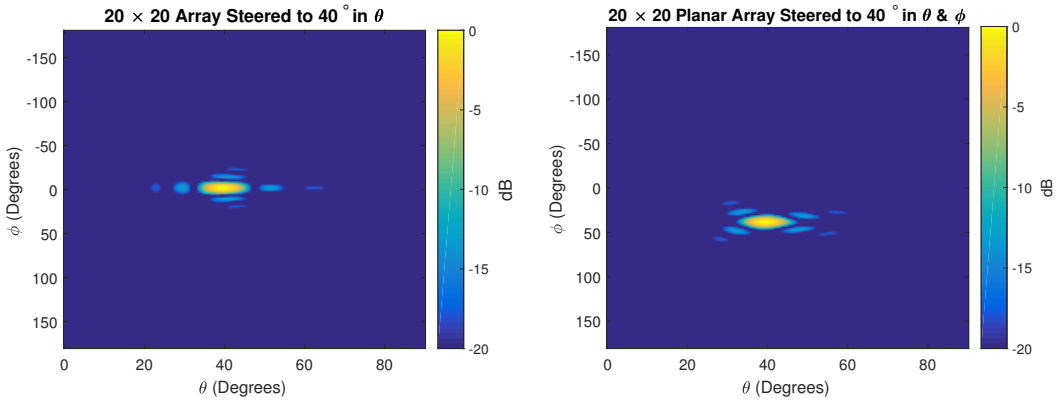


Figure 42: Sudoku interleaved array compared to a uniform array quadrant layout.

The results of the Sudoku and random permutation for a single main beam are shown in Fig. 44. The results are normalized to the ideal cases in Fig. 43, and therefore will have a maximum value lower than 0 dB due to the fraction of elements used. We observe both the Sudoku and random based steering result in very similar array factors. The main beam is approximately 12 dB lower than the ideal case and has sidelobes mostly 10 dB below the peak value.



(a) (b)

Figure 43: 20×20 Planar array using all elements: (a) Main beam steered to 40° in the θ direction; (b) Main beam steered to 40° in both the θ and ϕ directions.

Next, we examine the scenario in which all of the elements are turned on. Therefore, the resulting array factor will have four main beams at the assigned phases given in Table 7 for the case of either the θ direction, or θ and ϕ directions. The results for the Sudoku and random multiple beam scenario are shown in Fig. 45. The random permutation is normalized to the Sudoku results: therefore, we see that the largest value in the random array is larger than 0 dB. While both scenarios show the main beams, the Sudoku based

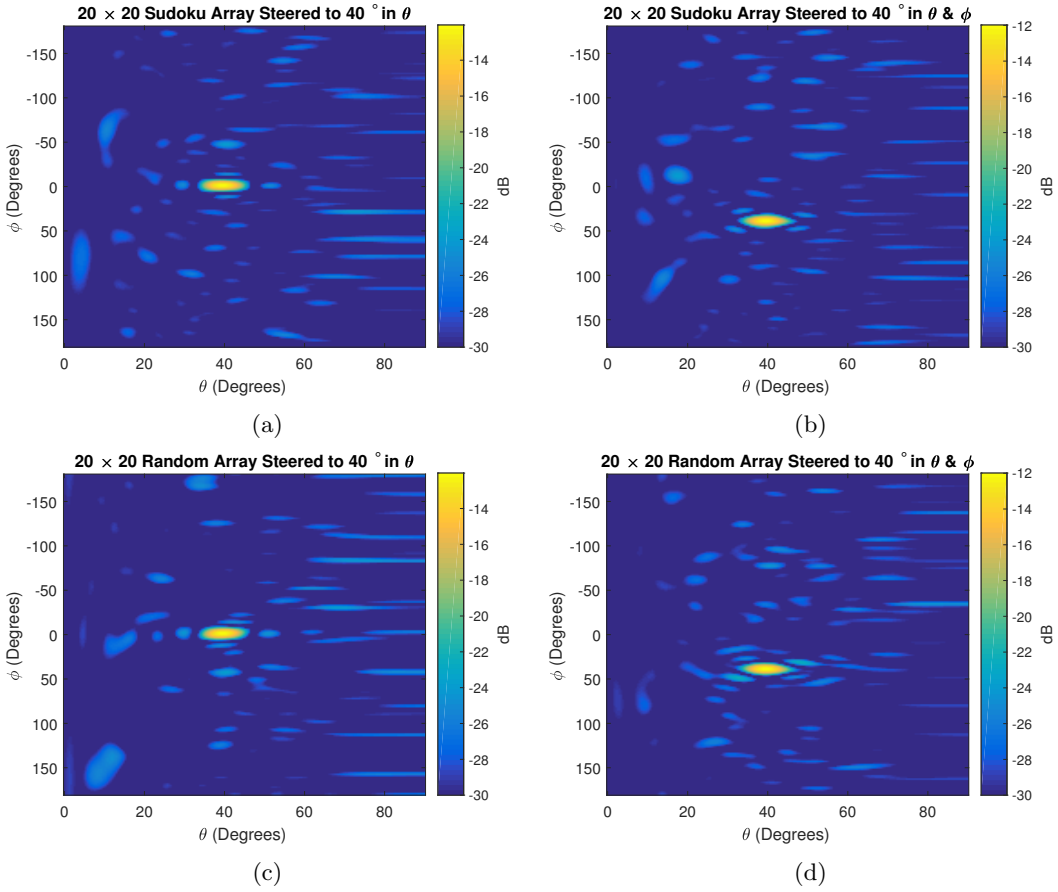


Figure 44: Single beam steering: (a) Sudoku based steering for main beam 40° in the θ direction; (b) Sudoku based steering for main beam 40° in both the θ and ϕ directions; (c) Random based steering for main beam 40° in the θ direction; (d) Random based steering for main beam 40° in both the θ and ϕ directions.

approach appears to have a more uniform amplitude across the different beams, while the random case has less uniformity across the beams.

We can examine this more closely in the Fig. 45e which shows a two-dimensional cross cut of the two array factors for $\phi = 0°$. We can see that indeed the random array does not provide as even an amplitude distribution across the four beams. Obviously using the random approach, the resulting beam amplitudes could change drastically from each iteration depending on how the element phases get assigned. The Sudoku interleaving approach based on the row, column, and box constraints provides a more even spread of the phases across the array so that a single area of the array is not dominated by a particular phase.

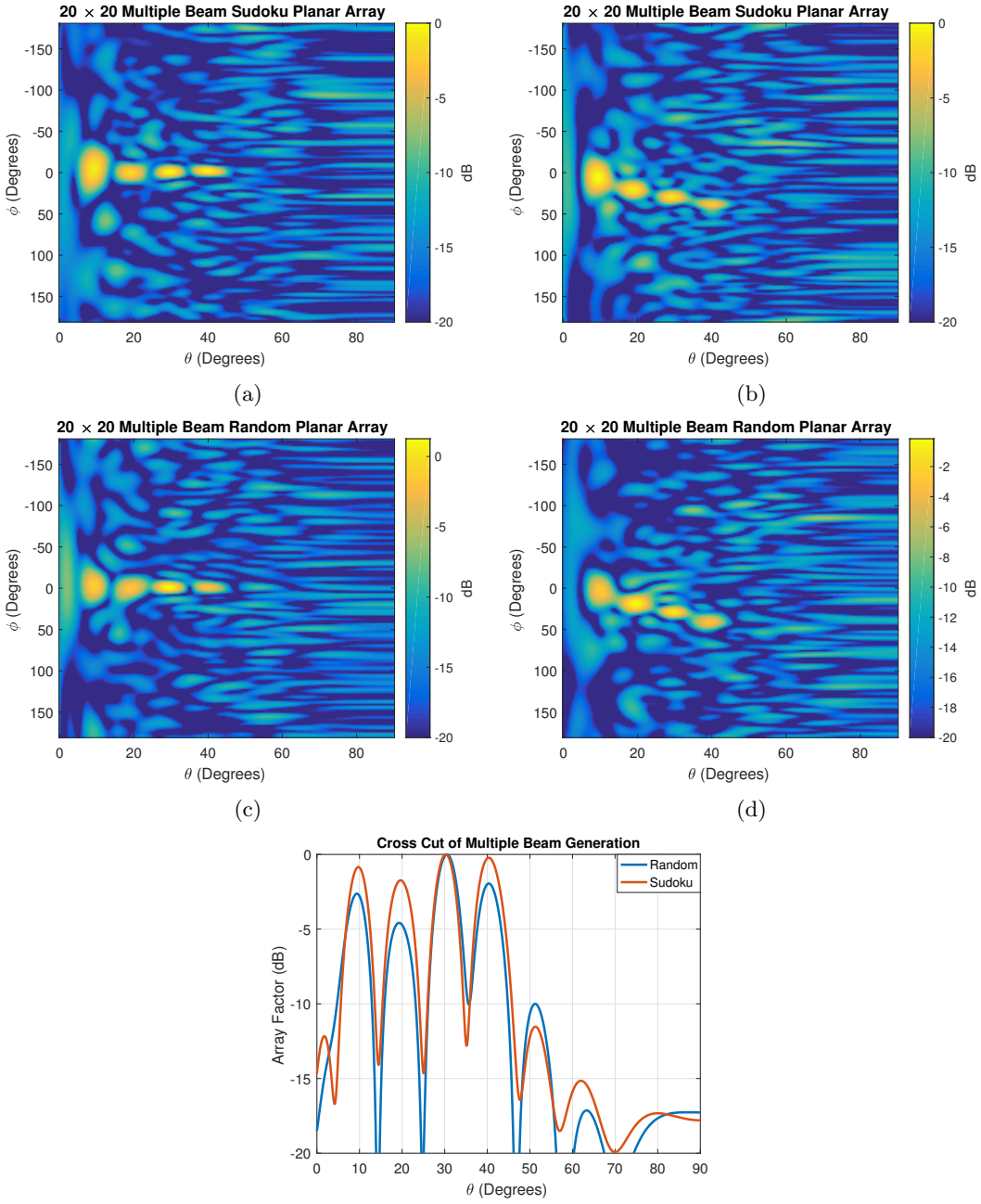


Figure 45: Multiple beam steering: (a) Sudoku based steering for beams 10°, 20°, 30°, 40° in the θ direction; (b) Sudoku based steering for main beams 10°, 20°, 30°, 40° in both the θ and ϕ directions; (c) Random based steering for main beams 10°, 20°, 30°, 40° in the θ directions; (d) Random based steering for main beams 10°, 20°, 30°, 40° in both the θ and ϕ directions; (e) Cross cut for $\phi = 0°$ for the Sudoku and random based arrays.

3.3 Sudoku Array Thinning

Since only a fraction of the elements were used during the investigation of array steering, it was only natural to extend the analysis to array thinning. Large antenna arrays have hundreds of elements where some of the elements can be turned off or removed without drastically altering the resulting array factor. The reduction of antenna elements can save cost, reduce complexity, and minimize system weight. Previous research in this area used optimization strategies such as genetic algorithms to optimize the removal of elements with respect to a desirable criteria such as sidelobe level [23]. We investigated the application of a Sudoku rule set so that the resulting array is judiciously thinned. We compared the Sudoku based approach using an array factor following random element removal. We used the same 20×20 antenna array comprised of 400 elements.

The Sudoku array thinning was compared to random thinning for thinned percentages of 25%, 50%, and 75% in Fig. 46 with the main beam scanned to 40°. The resulting thinned arrays are normalized with respect to the maximum of the ideal array factor in Fig. 43a. The results are shown as two-dimensional images with ϕ on the y-axis ranging from $[-180, 180]$ degrees and θ on the x-axis ranging from $[0, 90]$ degrees.

Generally the results are quite comparable between the random and Sudoku based thinning. The array factor still displays low sidelobe levels of 20 dB when 25% of the elements are removed. As more and more elements are turned off to achieve thinning percentages of 50% and 75% the side lobes quickly rise and the array factor is noticeable degraded; however it appears that the Sudoku thinning shows the sidelobes more clearly distributed around the main beam pointing direction. Comparing the thinned arrays to the ideal case, the main beam amplitude clearly lowers as the elements are removed with 25% of the elements causing approximately a 12-dB loss.

Observing Fig. 47a, we show a slice of the thinned Sudoku array factor in the XZ plane ($\phi = 0°$). We make a comparison in Fig. 47b between an ideal array, represented by the blue line, in which all of the elements are turned on and when the array is thinned to 50%, for both Sudoku and random based arrays. The Sudoku and random based arrays are normalized to the maximum of the ideal array, resulting in lower amplitude compared to the ideal array factor. We can clearly see that the Sudoku thinning, represented by the red line, has the exactly the same array factor as an array that has all of the elements turned on, whereas the random thinning, represented by the yellow line, causes the sidelobes around the main beam to fluctuate and deviate from the ideal array factor.

In order to see why Sudoku thinning keeps the same array side-lobes as an ideal array in the XZ or YZ planes, we examine Fig. 48 which shows a 4×4 Sudoku matrix with only the 1s present; the rest of the elements are assigned a 0 corresponding to an element that is turned off. Thus, when applying the planar array factor given previously in equation (11), the 0s do not contribute to the array factor, while only the 1s do. The result of evaluating the summation for the XZ plane ($\phi = 0°$; thus $\cos\phi = 1$ and $\sin\phi = 0$) assuming equal element spacing in the x and y directions, $\theta_0 = \phi_0 = 0$, and the current amplitudes I_{mn} equal to unity, results in,

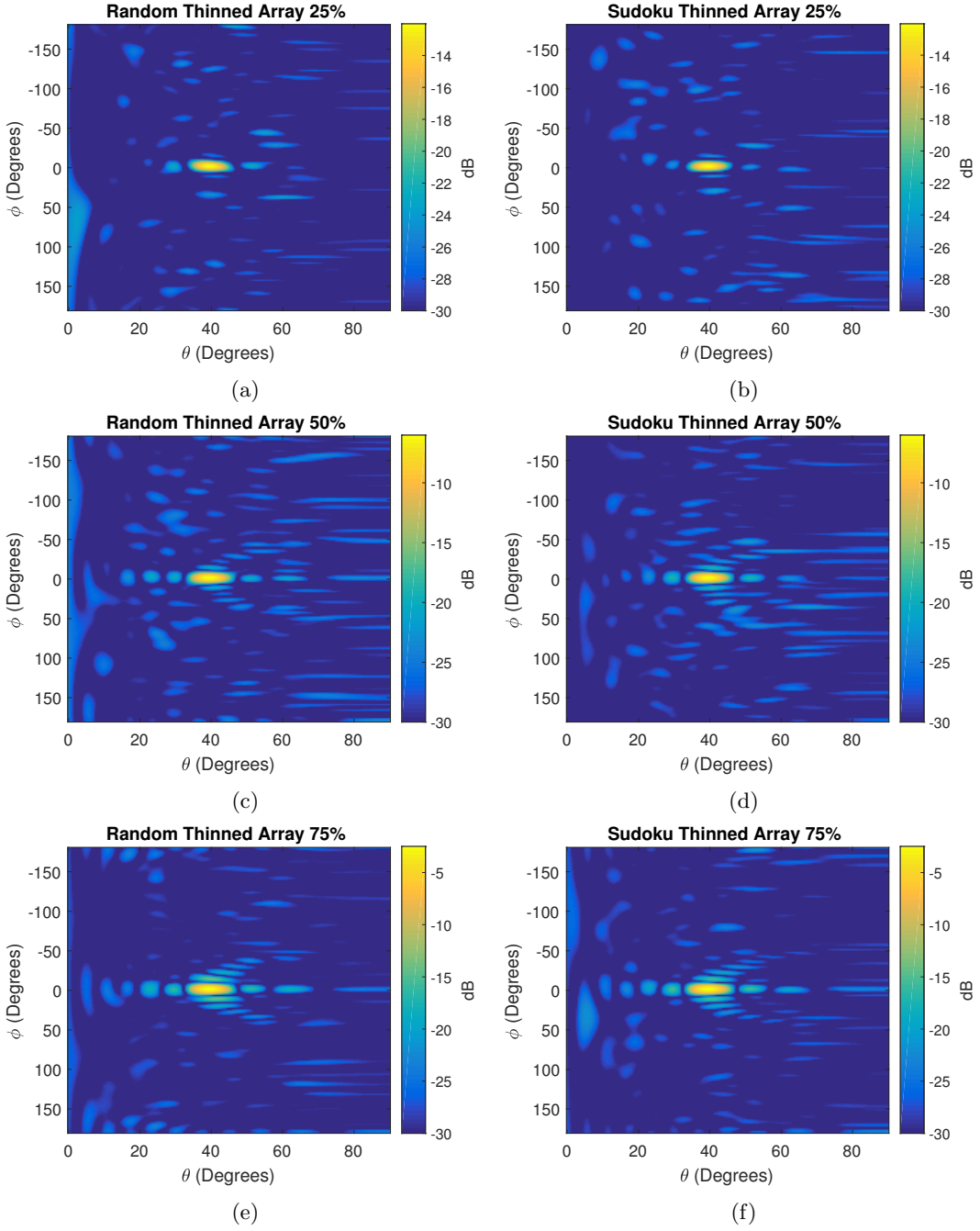
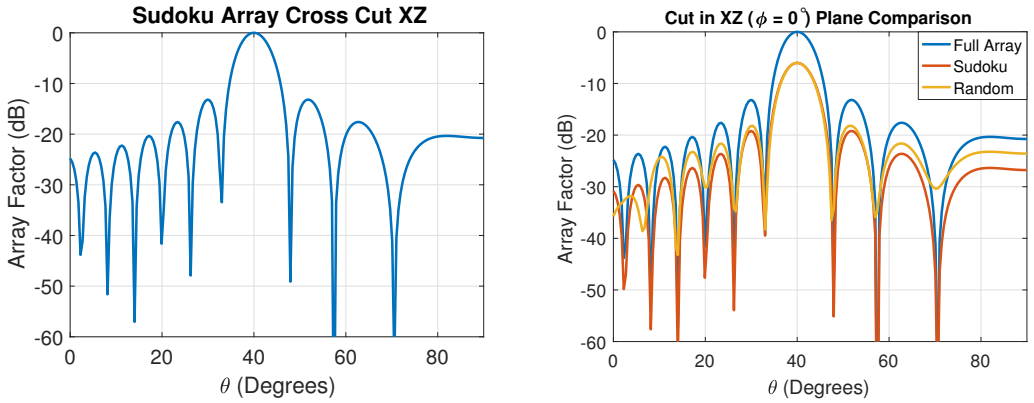


Figure 46: Thinned 20×20 planar array using random and Sudoku based thinning. (a) random thinning of 25%, (b) Sudoku thinning of 25%, (c) random thinning of 50%, (d) Sudoku thinning of 50%, (e) random thinning of 75%, (f) Sudoku thinning of 75%.



(a) (b)

Figure 47: Cross cut of the scanned 20×20 planar array: (a) Thinned Sudoku array cross cut in the XZ plane; (b) Comparison of the Sudoku and randomly thinned array factor to that of an array with all of the elements turned on.

$$\begin{aligned} AF(\theta,\phi) &= 1 + \mathrm{e}^{j\beta d \sin\theta[\cos\phi+3\sin\phi]} + \mathrm{e}^{j\beta d \sin\theta[2\cos\phi+2\sin\phi]} + \mathrm{e}^{j\beta d \sin\theta[3\cos\phi+\sin\phi]}, \\ &= 1 + \mathrm{e}^{j\beta d \sin\theta} + \mathrm{e}^{j2\beta d \sin\theta} + \mathrm{e}^{j3\beta d \sin\theta}. \end{aligned} \tag{12}$$

The resulting array factor in equation (12) clearly represents that of a linear array. A similar result is apparent when evaluating the array factor in the YZ ($\phi = 90°$) plane. In other words, Sudoku or Latin Square based thinning removes an element from every row and column resulting in removal of linear array products.

One might be interested in what advantages, if any, Sudoku based thinning offers over random thinning, or over more robust optimization methods. We believe that Sudoku thinning offers a more uniform thinning of the array since the row, column, and subgrid Sudoku constraints causes thinning to not be concentrated in one area, as what could happen in randomly thinning the array. Furthermore, we see good sidelobe behavior within the thinned Sudoku arrays as well as the cardinal cuts holding the ideal planar array shape.

$$\left[\begin{array}{cc|cc} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{array}\right]$$

Figure 48: 4×4 Sudoku array with only the 1s present to study the effects of turning off elements with row, column, and subgrid constraints.

3.4 Random Planar Arrays

Owing to the equal element spacing, the periodic arrays are susceptible to grating lobes when the spacing becomes too large for a particular scan angle. Using irregular spacing has the benefit of reducing the appearance of grating lobes while also increasing the bandwidth of the antenna array.

3.4.1 Perturbed Planar Arrays

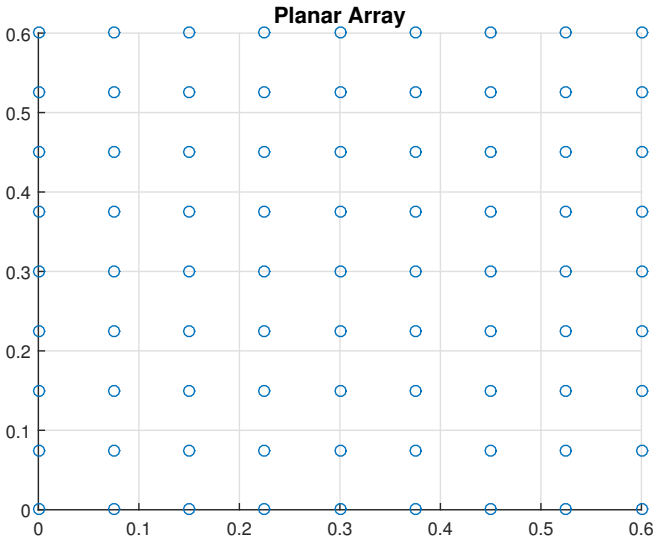
We investigate using Sudoku puzzles as perturbations within the planar array of the same size. This is accomplished by modifying the planar array equation (11) so that d_x and d_y have an added perturbation δ. The spacing in the X and Y directions then become $d_x = d_x + \delta$ and $d_y = d_y + \delta$, respectively. The degree of perturbation comes from the Sudoku puzzle.

First, let us assume we have a 9×9 planar array with $\lambda_0 = 15$ cm corresponding to a frequency of 2 GHz along with equal amplitude excitation of 1 and uniform spacing of $\lambda_0/2$ as shown in Fig. 49a. Generation of two random Sudoku matrices, one for the x-direction, and the other for the y-direction will serve as the perturbations. We would like to add perturbations based upon the bandwidth we want the array to operate over; in this case, we consider an operating frequency in the range 1-3 GHz. The bandwidth of 2 GHz was divided evenly among the 9 Sudoku numbers giving nine frequencies spaced 250-MHz apart. The corresponding wavelengths, in cm, were

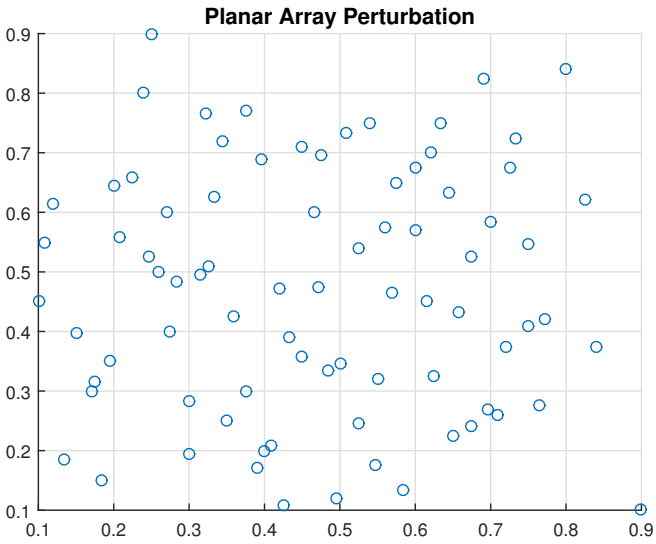
$$\lambda = [0.3,\ 0.24,\ 0.2,\ 0.1714,\ 0.1333,\ 0.12,\ 0.109, 0.1].$$

The number 1 corresponding to $\lambda = 0.3$ cm ($f = 1$ GHz) and the number 9 corresponding to $\lambda = 0.1$ cm ($f = 3$ GHz) with the rest of the integers following in decreasing order in wavelength or increasing order in frequency. Utilizing the randomly generated Sudoku matrices, the numbers were replaced with their respective perturbation values and added to each element during the array factor construction. The output of this procedure is shown in Fig. 49b where we now see the resulting array has a random like structure, while also increasing in size by $2\lambda_0$.

The size of the array has increased from $0.6\lambda \times 0.6\lambda$ to $0.9\lambda \times 0.9\lambda$. We compare the resulting perturbed array to the equally spaced planar array for the frequencies of 1, 2, and 3 GHz in Fig. 50 for $\phi = 0^\circ$ and $\phi = 90^\circ$. We clearly see lower sidelobes in the perturbed array. This suggests that the Sudoku based perturbations are able to achieve good random arrays resulting in better sidelobe performance.

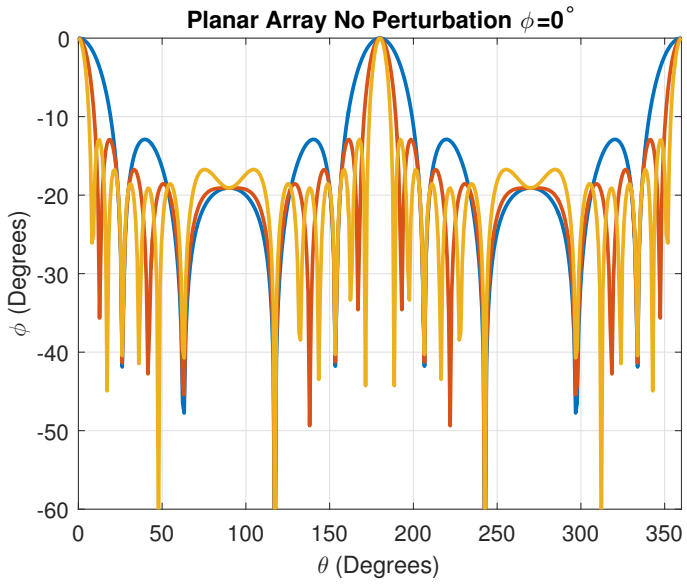


(a)

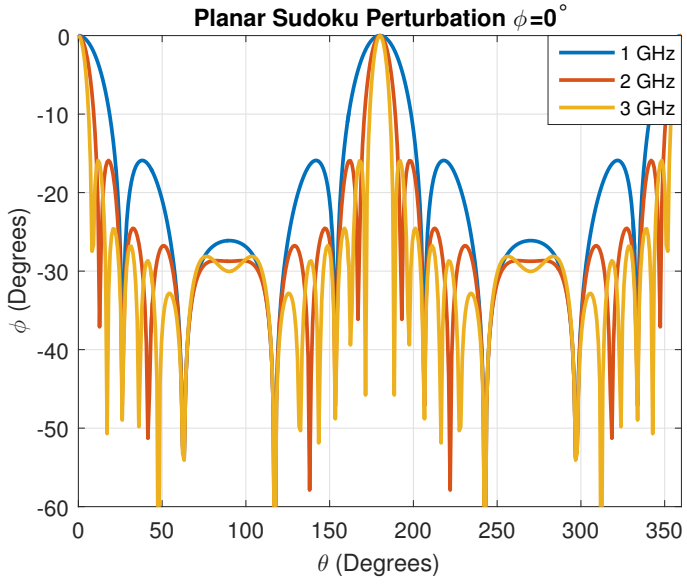


(b)

Figure 49: Array topology (a) Planar array with $\lambda/2$ spacing corresponding to 2 GHz, (b) Planar array with perturbations.

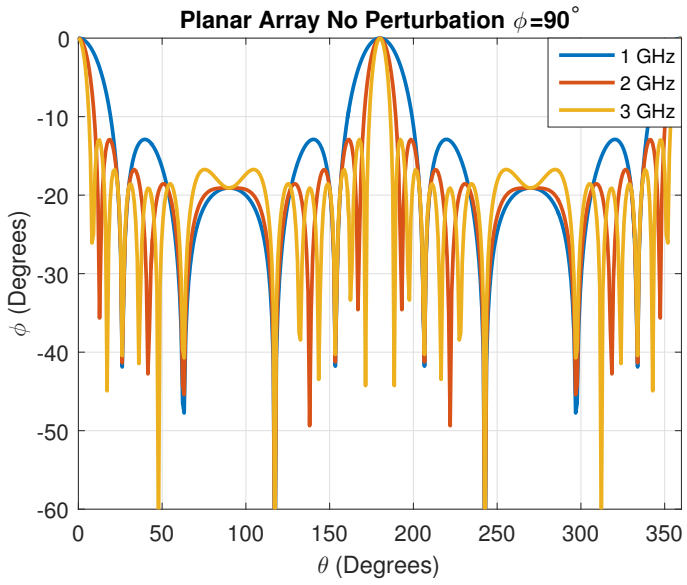


(a)

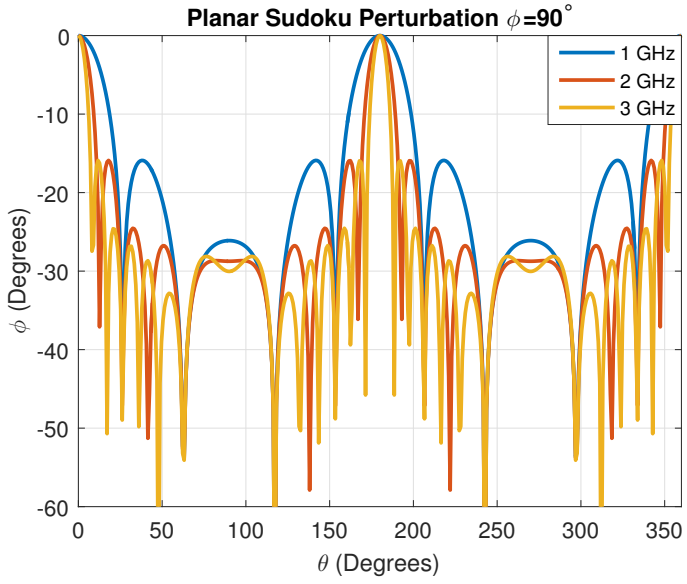


(b)

Figure 50: Planar array with spacing equal to the center frequency of 2 GHz and $\phi = 0°$. (a) No perturbation, (b) Sudoku perturbation.



(a)



(b)

Figure 51: Planar array with spacing equal to the center frequency of 2 GHz and $\phi = 90°$. (a) No perturbation, (b) Sudoku perturbation.

4 Radar Target Detection Simulations

Simulated radar target scenarios for single targets were implemented using Sudoku frequency codes. For the frequency hopped sequence, the transmitted signal $x(t)$, is given by [24]

$$x(t) = \frac{1}{\sqrt{NT_c}} \sum_{n=1}^{M} u(t - nT_c), \tag{13}$$

where N is the number of subpulses and T_c is the subpulse length. The complex envelope, $u(t)$, is defined as

$$u(t) = \begin{cases} \exp(j2\pi f_n t) & , 0 \leq t \leq T_c \\ 0 & , \text{elsewhere.} \end{cases} \tag{14}$$

The frequency of the subpulse F_n is given by

$$F_n = F_0 + S_n \Delta_f, \tag{15}$$

where F_0 is the carrier frequency, Δf is the frequency spacing of the hopped waveform given by $1/Tc$, and S is the frequency hopped sequence, represented as

$$S = [a_1,\ a_2,\ a_3\ ...a_M]. \tag{16}$$

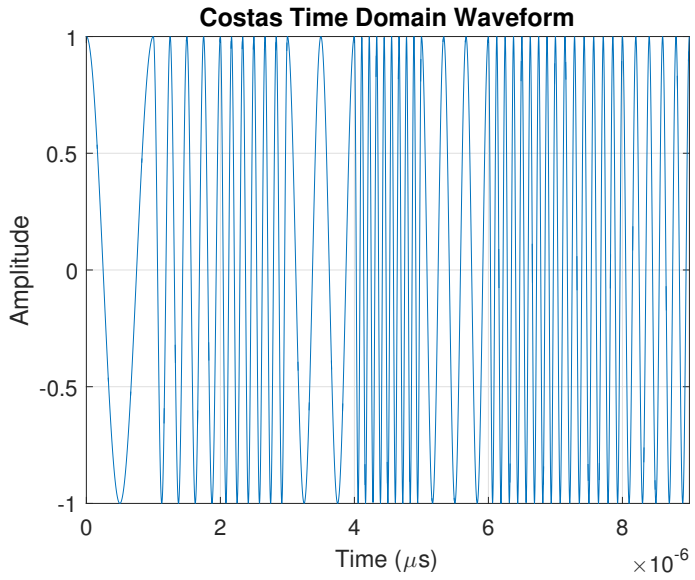
Before discussing the Sudoku, we first look at a Costas frequency sequence of length 9. The simulated target and radar characteristics for the simulations are given in Table 8 where we use a bandwidth of 9 MHz, a carrier frequency of 10 GHz, and pulse width of 1 μs for a total contiguous length of 9 μs.

Table 8: Radar Costas Simulated Scenario Parameters.

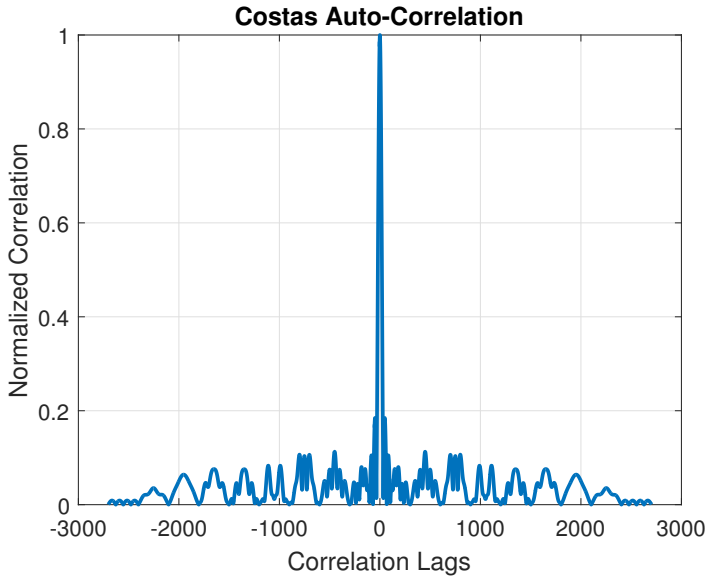
Radar Parameter	Values
Code Length	9
T_c	1 μs
F_0	10 GHz
Δf	$1/T_c$
Bandwidth	$N\Delta f = 9$ MHz
Target Range	2 km
Target RCS	10 m^2

The transmitted Costas waveform is given in Fig. 52a where it is easy to see the arrangement of the frequencies are in a non-linear order. The auto-correlation of the transmit waveform is shown in Fig. 52b where we see the pulsed compressed response. A Costas waveform has a pulse compression of N^2; in this case, with a length 9 sequence, the

compression ratio is 81. The resulting spectrum of the transmit waveform is shown in Fig. 53 showing the 9 MHz of bandwidth.

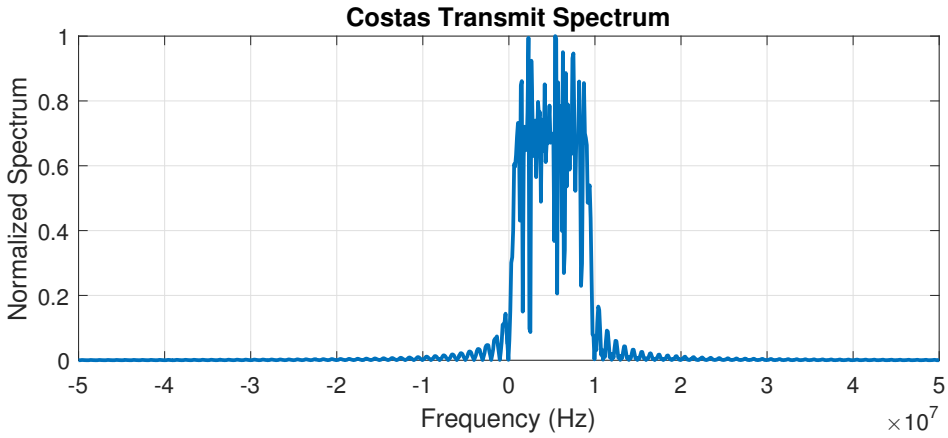


(a)



(b)

Figure 52: (a) Transmitted Costas waveform of length 9 with carrier frequency removed, (b) Normalized auto correlation for the transmit waveform of a Costas frequency code of length 9.

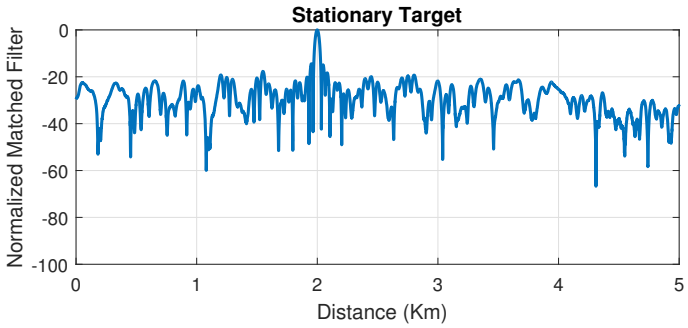


(a)

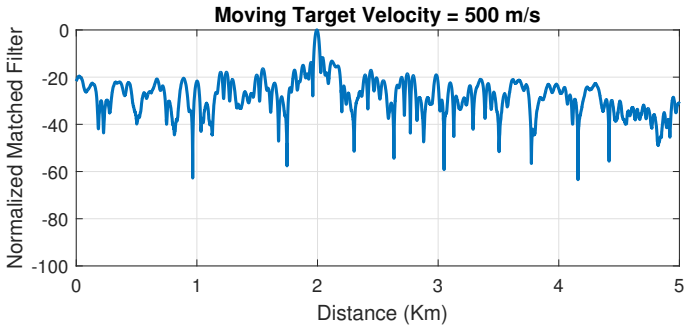
Figure 53: Spectrum of transmit waveform showing the 9 MHz bandwidth.

Using the previously described Costas waveform, we perform different radar simulations for single and multiple target scenarios as well as compare the results for stationary and moving targets. The matched filter outputs for a single target is shown in Fig. 54 for the stationary case and when the target is moving. Examining Fig. 54a, we see the correct peak at 2 km. As the speed of the target is changed to 500 m/s in Fig. 54b and 1000 m/s in Fig. 54c, we see the peak correlation lower and become less correlated. This is very evident when the velocity is 1000 m/s which now shows that the peak is spread more in range resulting in ambiguity as to the exact target location.

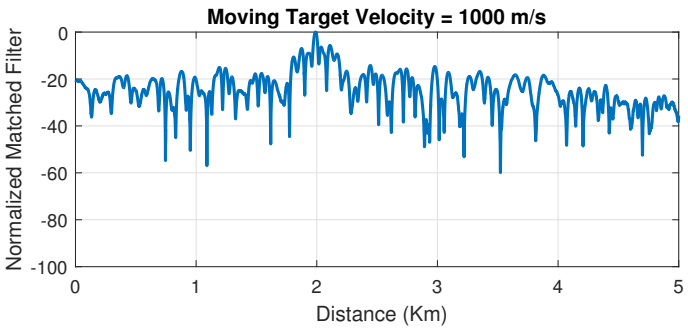
Multiple targets are now simulated in Fig. 55 with the targets located at 2 km and 2.3 km. The stationary targets result is shown in Fig. 55a where we clearly see the two correlation peaks corresponding to the targets. Due the thumbtack like nature of the Costas sequence, we have low range sidelobes resulting is less ambiguity for closely spaced targets. The two targets are next simulated at speeds of 500 m/s in Fig. 55b and 1000 m/s in Fig. 55c. When the targets are moving, the resulting matched filter does not clearly show multiple targets are within the scene. Clearly, the Costas type of waveforms are not Doppler tolerant as observed from their previously discussed ambiguity functions.



(a)

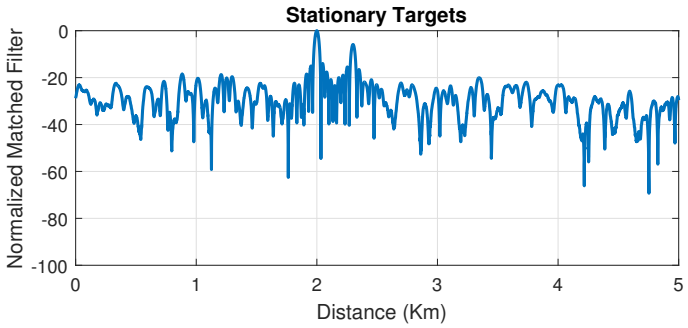


(b)

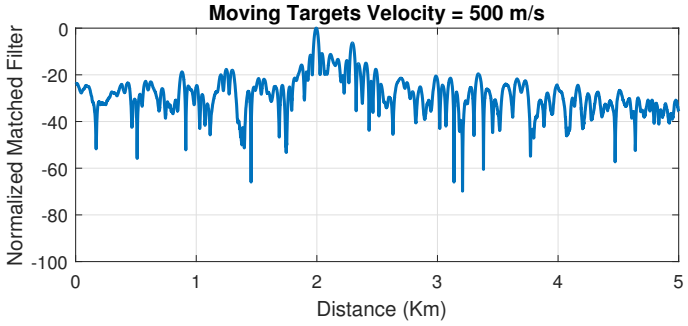


(c)

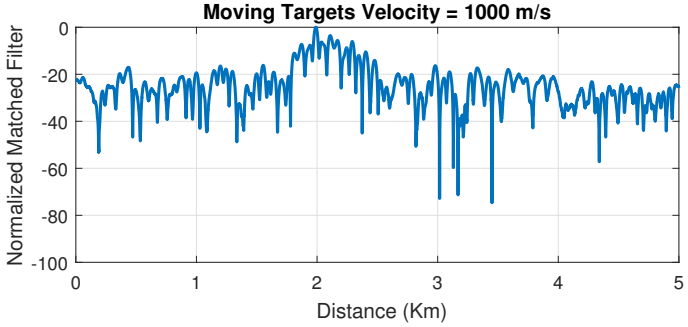
Figure 54: (a) Matched filter output for a length 9 Costas sequence and a single stationary target located 2 km in the range direction, (b) Matched filter output for a length 9 Costas sequence and a single stationary target located 2 km in the range direction with a speed of 500 m/s, (c) Matched filter output for a length 9 Costas sequence and a single stationary target located 2 km in the range direction with a speed of 500 m/s.



(a)



(b)



(c)

Figure 55: (a) Matched filter output for a length 9 Costas sequence and a two stationary target located 2 km and 2.3 km in the range direction, (b) Matched filter output for a length 9 Costas sequence and a two moving targets located 2 km and 2.3 km in the range direction with a speed of 1000 m/s, (c) Matched filter output for a length 9 Costas sequence and a two moving targets located 2 km and 2.3 km in the range direction with a speed of 1000 m/s.

4.1 Sudoku Radar Simulations

We perform radar simulations for single targets with various Sudoku frequency code lengths of 9, 16, 20, and 81. As the codes get longer, the range resolution naturally improves. The radar parameters for the Sudoku simulations are given in Table 9 where the pulse length and carrier frequency stay constant and the bandwidth is varied based upon the code lengths.

Table 9: Radar simulated scenario parameters.

Radar Parameter	Values
Code Length	9, 16, 21, 81
T_c	1 μs
F_0	10 GHz
Δf	$1/T_c$
Bandwidth	9 MHz, 16 MHz, 21 MHz, 81 MHz
Target Range	2 km
Target RCS	10 m^2

4.2 Single Target Scenario

The results in Fig. 56 show a stationary target with the various code lengths. Viewing the results show that the longer codes provide good pulse suppression similar to that of Costas frequency hopped waveforms. Introducing target velocity into the simulations the single moving target results are shown in Fig. 57. Just like the Costas frequency hopped waveforms, the Sudoku hopped sequences are not Doppler tolerant, matching up with their ambiguity function results.

We set the target velocity the same for each of the Sudoku code lengths. Looking back at the ambiguity function plots, we know that the Doppler axis is normalized as given in equation (17), where δf is the Doppler shift integer value. We can solve for the Doppler shift F_d and solve for the corresponding velocity using equation (18).

$$F_d N T_c = \delta f \tag{17}$$

$$v = \frac{\lambda F_d}{2} \tag{18}$$

The resulting velocity depends upon the code length, frequency, and subpulse length. It is easy to see that as the code length gets longer, the velocity will also be smaller to represent the same Doppler shift. This should make intuitive sense since the longer codes provide a better approximation to the thumbtack like response. We can see this clearly when comparing the length 81 Sudoku sequence in Fig. 57d and Fig. 57e where the velocity is set to only 200 m/s and has nearly the same matched filter response.

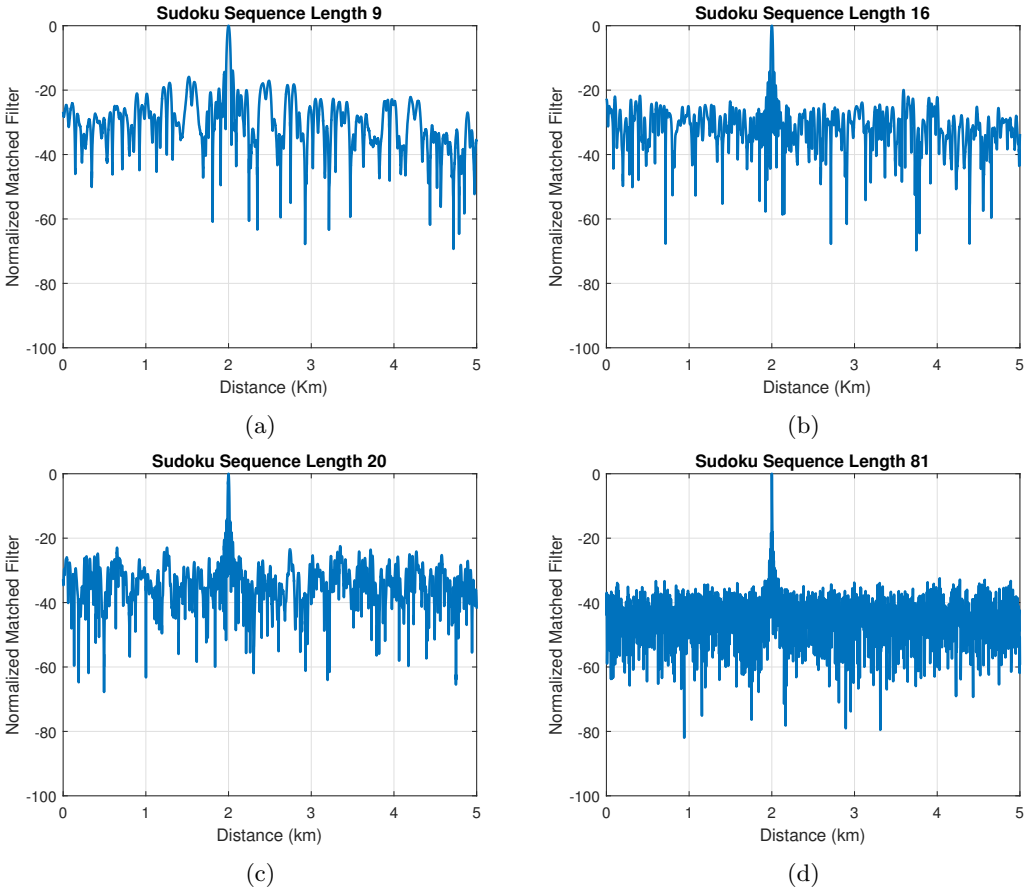


Figure 56: (a) Matched filter output for a length 9 Sudoku sequence and a single stationary target located 2 km in the range direction, (b) Matched filter output for a length 16 Sudoku sequence and a single stationary target located 2 km in the range direction, (c) Matched filter output for a length 20 Sudoku sequence and a single stationary target located 2 km in the range direction, (d) Matched filter output for a length 81 Sudoku sequence and a single stationary target located 2 km in the range direction.

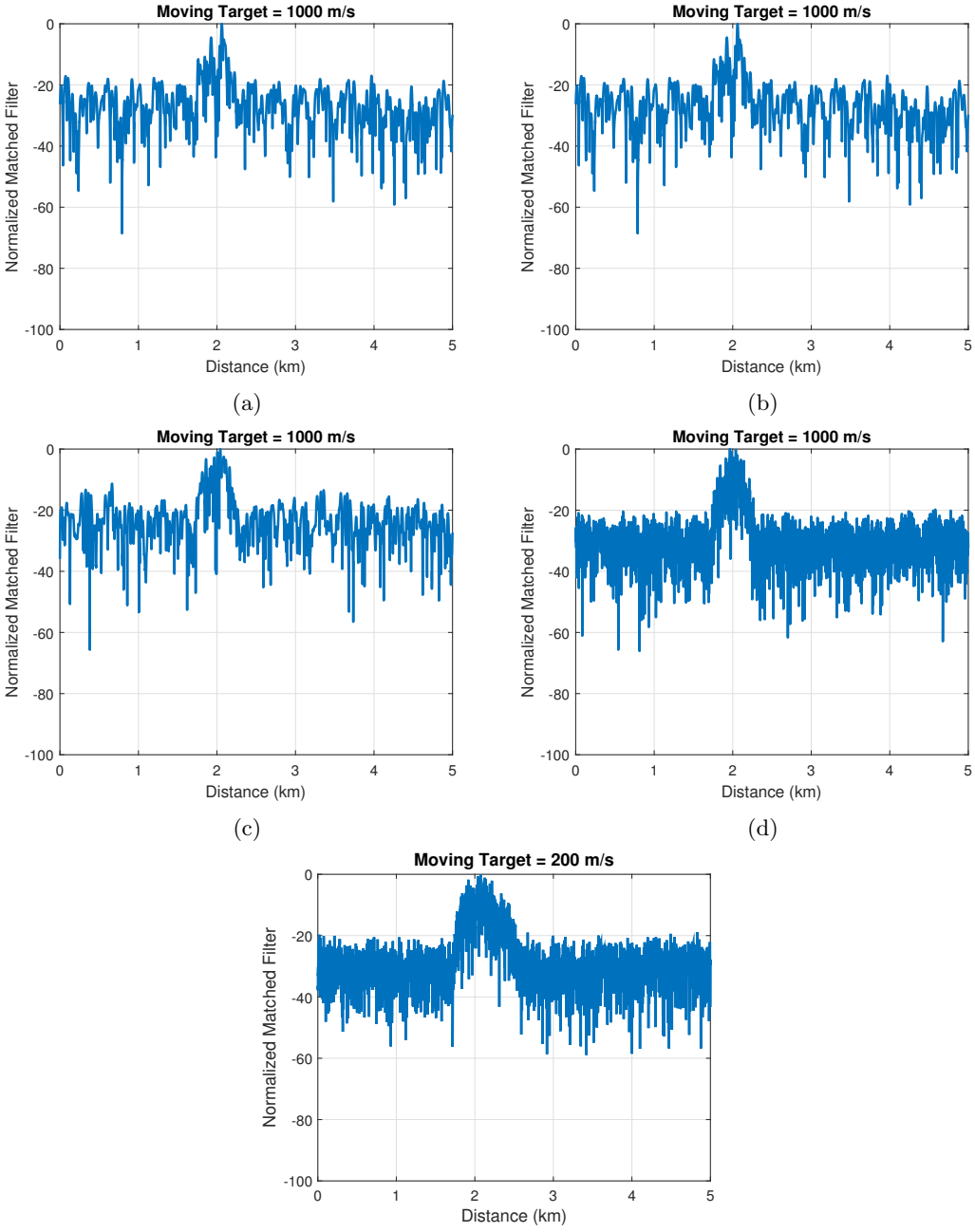


Figure 57: (a) Matched filter output for a length 9 Sudoku sequence with moving target located 2 km and velocity of 1000 m/s, (b) Matched filter output for a length 16 Sudoku sequence with moving target located 2 km and velocity of 1000 m/s, (c) Matched filter output for a length 20 Sudoku sequence with moving target located 2 km and velocity of 1000 m/s, (d) Matched filter output for a length 81 Sudoku sequence with moving target located 2 km and velocity of 1000 m/s, (e) Matched filter output for a length 81 Sudoku sequence with moving target located 2 km and velocity of 200 m/s.

4.3 Sudoku Multiple Target Scenario

Multiple targets located at 2 km and 2.3 km are now simulated in Fig. 58 for the stationary case, and Fig. 59 for the moving target scenario. The stationary targets result clearly shows the two correlation peaks corresponding to the two targets. Due the thumbtack like nature of the Sudoku sequence, we have low range sidelobes resulting is less ambiguity for closely spaced targets. Next, the two targets are simulated with a velocity of 1000 m/s. When the targets are moving, the resulting matched filter output does not clearly show multiple targets are within the scene. Clearly, the Sudoku type of waveforms are not Doppler tolerant as observed from their previously discussed ambiguity functions.

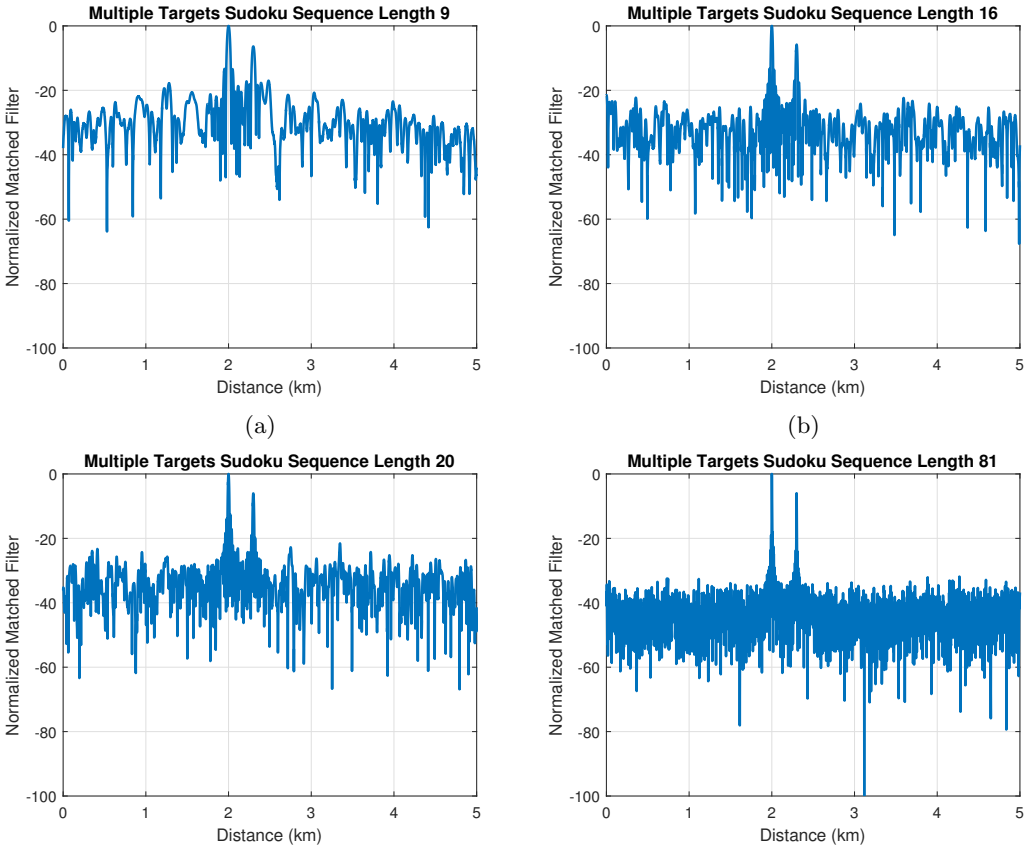


Figure 58: Matched filter output for a length 9 Sudoku sequence and a single stationary target located 2 km in the range direction, (b) Matched filter output for a length 16 Sudoku sequence and a single stationary target located 2 km in the range direction, (c) Matched filter output for a length 20 Sudoku sequence and a single stationary target located 2 km in the range direction, (d) Matched filter output for a length 81 Sudoku sequence and a single stationary target located 2 km in the range direction.

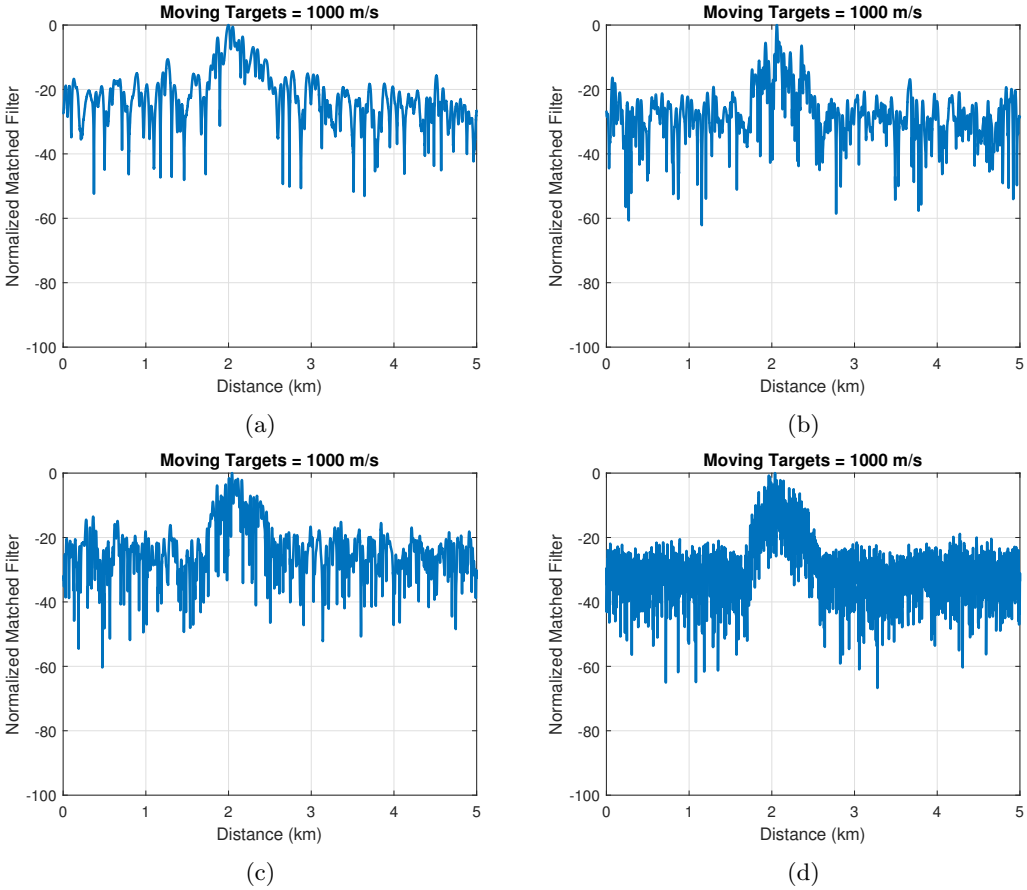


Figure 59: Matched filter output for a length 9 Sudoku sequence with moving targets located at 2 km and 2.3 km and velocity of 1000 m/s, (b) Matched filter output for a length 16 Sudoku sequence with moving targets at 2 km and 2.3 km and velocity of 1000 m/s, (c) Matched filter output for a length 20 Sudoku sequence with moving targets at 2 km and 2.3 km and velocity of 1000 m/s, (d) Matched filter output for a length 81 Sudoku sequence with moving targets at 2 km and 2.3 km and velocity of 1000 m/s.

5 Summary and Future Direction

This report examined Sudoku puzzles for various radar applications. The majority of the work examined Sudoku for discrete frequency coded waveforms in similar fashion to the well researched Costas coded sequences. Costas sequences have the property of approaching the ideal "thumbtack" like response when evaluating the ambiguity function. The thumbtack response is due to the fact that the number of collisions using a Costas sequence is limited to at most one for a given shift in time or frequency. A Costas sequence length determines the sharpness of the response in the delay and Doppler planes, with longer codes producing a narrower response. In regards to High resolution radar, waveforms with little to no ambiguities in the delay direction are desirable as they will not overwhelm weaker targets.

Sudoku puzzles have row, column, and sub grid constraints that make them attractive to analyze in regards to frequency coded waveforms. We found that, in general, the Sudoku puzzle codes perform well and approach the thumbtack like response in a similar vein to Costas codes. Sudoku puzzles as they increase in size start to show an increase in the number of collisions but still perform well as shown in the ambiguity function plots presented in this report. An advantage of using Sudoku codes for frequency coding is the large number of puzzles that can be easily generated compared to the finite number of Costas codes for a given order. It has been proved that there are approximately 6.671×10^{21} valid Sudoku grids of size 9×9 [25], with this number increasing with the order of the Sudoku puzzle. Additionally, it was noted that certain arrangements of Sudoku puzzles are in fact Costas coded solutions where different geometric operations of rotation and reflection could potentially be applied to fill out the Sudoku matrix.

Radar simulations for single and multi target scenarios were undertaken to evaluate the performance of Costas and Sudoku coded waveforms. We examined several lengths of Sudoku codes with stationary and moving targets noting how the results showed low range ambiguities for separation of closely spaced targets. However, the introduction of velocity clearly showed how the target response is severely degraded providing low correlation with the transmit waveform. Doppler also has the effect of reducing the ability to find closely spaced targets through decorrelation causing the matched filter output to deviate from the ideal thumbtack response.

Lastly, the Sudoku puzzles were investigated for techniques in planar array analysis. Specifically, applications in random spacing, array thinning, and array interleaving. Periodic arrays have grating lobes that appear when the element spacing is greater than a certain threshold, using irregular or random spacing prevents grating lobes from appearing while allowing the geometry to operate over many frequencies. In regards to array thinning, often large antenna arrays suffer little degradation when small percentages of elements are removed or fail. Thinning through the use of Sudoku puzzles helps to apply a uniform thinning across the array. Secondly, Sudoku array thinning leaves the cardinal planes looking like a full antenna array since Sudoku thinning removes linear array products, while randomly removing elements does not. Furthermore, the use

of Sudoku puzzles were used for interleaving multiple arrays, which proved beneficial over randomly assigning the phases to the array by providing a more uniform amplitude across the multiple beams.

The next step is looking at future research into Sudoku puzzles applications. Here, we present some topics that seem to have potential for Sudoku. One such area is steganography which is the study of hiding information. Due to the randomness and constraints of Sudoku puzzles, information could be scrambled before being transmitted over a medium. Prior work in steganography has mainly been done in image processing as well as speech analysis. Similarly, Sudoku puzzles can be applied to cryptography where the adversary knows the data are encrypted but cannot decipher the information. Additional work is phase coding or amplitude coding seems also a likely candidate for research. Added work in phase arrays is also viable such as investigating other types of array geometries such as circular arrays.

References

[1] J. Taylor, *Ultrawideband Radar: Applications and Design.* CRC Press, 2016.

[2] A. D. Keedwell and J. Dénes, *Latin squares and their applications.* Elsevier, 2015.

[3] J.-P. Delahaye, "The science behind sudoku," *Scientific American*, vol. 294, no. 6, pp. 80–87, 2006.

[4] P. K. Newton and S. A. DeSalvo, "The Shannon entropy of Sudoku matrices," in *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 466, pp. 1957–1975, The Royal Society, 2010.

[5] R. M. Pedersen and T. L. Vis, "Sets of mutually orthogonal Sudoku latin squares," *The College Mathematics Journal*, vol. 40, no. 3, pp. 174–181, 2009.

[6] A. Keedwell, "Constructions of complete sets of orthogonal diagonal sudoku squares," *Australas. J. Combin*, vol. 47, pp. 227–238, 2010.

[7] J. Sarkar and B. K. Sinha, "Sudoku squares as experimental designs," *Resonance*, vol. 20, no. 9, pp. 788–802, 2015.

[8] C.-C. Chang, Y.-C. Chou, and T. D. Kieu, "An information hiding scheme using Sudoku," in *Innovative Computing Information and Control, 2008. ICICIC'08.*, pp. 17–17, IEEE, 2008, doi: 10.1109/ICICIC.2008.149.

[9] Y. Wu, Y. Zhou, S. Agaian, and J. P. Noonan, "2d Sudoku associated bijections for image scrambling," *Information Sciences*, vol. 327, pp. 91–109, 2016.

[10] P. K. Newton and S. A. DeSalvo, "The shannon entropy of sudoku matrices," in *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 466, pp. 1957–1975, The Royal Society, 2010.

[11] H. Urkowitz, C. Hauer, and J. Koval, "Generalized resolution in radar systems," *Proceedings of the IRE*, vol. 50, no. 10, pp. 2093–2105, 1962.

[12] A. W. Rihaczek, "Radar signal design for target resolution," *Proceedings of the IEEE*, vol. 53, no. 2, pp. 116–128, 1965.

[13] P. M. Woodward, *Probability and Information Theory, with Applications to Radar: International Series of Monographs on Electronics and Instrumentation*, vol. 3. Elsevier, 2014.

[14] M. Skolnik, *Introduction to Radar Systems.* Mc-Graw Hill Book Company, 1980.

[15] D. Wehner, *High Resolution Radar.* Artech House, 1987.

[16] N. Levanon, *Radar Principles.* Wiley-Interscience, 1988.

[17] J. P. Costas, "A study of a class of detection waveforms having nearly ideal range-doppler ambiguity properties," *Proceedings of the IEEE*, vol. 72, no. 8, pp. 996–1009, 1984.

[18] S. Maric, I. Seskar, and E. Titlebaum, "On cross-ambiguity properties of welch-costas arrays," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 30, no. 4, pp. 1063–1071, 1994.

[19] J. P. Costas, "A study of a class of detection waveforms having nearly ideal range doppler ambiguity properties," *Proceedings of the IEEE*, vol. 72, pp. 996–1009, Aug 1984.

[20] E. W. Kang, *Radar System Analysis, Design, and Simulation.* Artech House, 2008.

[21] W. L. Stutzman and G. A. Thiele, *Antenna Theory and Design.* John Wiley & Sons, 2012.

[22] R. L. Haupt, *Antenna Arrays: A Computational Approach.* John Wiley & Sons, 2010.

[23] R. L. Haupt, "Thinned arrays using genetic algorithms," *IEEE Transactions on Antennas and Propagation*, vol. 42, no. 7, pp. 993–999, 1994.

[24] N. Levanon and E. Mozeson, *Radar Signals.* John Wiley & Sons, 2004.

[25] B. Felgenhauer and F. Jarvis, "Enumerating possible sudoku grids," *Preprint available at http://www. afjarvis. staff. shef. ac. uk/sudoku/sudoku. pdf*, 2005.

Publications

Journal Papers:

1. T.D. Bufler, R. M. Narayanan, and K.D. Sherbondy, "Sudoku Inspired Designs for Radar Waveforms and Antenna Arrays," *Electronics* (special issue on Radio and Radar Signal Processing), submitted December 2016, under review.

Conference Papers:

1. T.D. Bufler, R. M. Narayanan, and K.D. Sherbondy, "Analysis of Sudoku Coded Waveforms and Application to Planar Phased Arrays," *Proc. IEEE International Radar Conference,* Seattle, WA, May 2017, submitted and under review.